



TUGAS AKHIR - TE141599

**PERANCANGAN KONTROLER *NEUROFUZZY* UNTUK
TELUSUR TRAJEKTORI PADA UAV *QUADCOPTER***

Hanief Sasongko
NRP 2214 105 035

Dosen Pembimbing
Ir. Rusdhianto Effendie A.K., M.T.
Mochammad Sahal, ST., M.Sc.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE141599

***DESIGN OF NEUROFUZZY CONTROLLER FOR
TRAJECTORY KEEPING OF THE UAV QUADCOPTER***

Hanief Sasongko
NRP 2214 105 035

Advisors

Ir. Rusdhianto Effendie A.K., M.T.
Mochammad Sahal, ST., M.Sc.

***ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017***

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “***Perancangan Kontroler Neurofuzzy Untuk Telusur Trajektori Pada UAV Quadcopter***” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 19 Januari 2017



Hanief Sasongko
NRP 2214 105 035

--halaman ini sengaja dikosongkan--

PERANCANGAN KONTROLER NEUROFUZZY UNTUK TELUSUR TRAJEKTORI PADA UAV QUADCOPTER

TUGAS AKHIR


**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**


**Pada
Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya**

Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II


Ir. Rusdhianto Effendie A.K., M.T.
NIP. 195704241985621001


Mocharsamad Sahal, ST., M.Sc.
NIP. 197011191998021002



--halaman ini sengaja dikosongkan--

PERANCANGAN KONTROLER *NEUROFUZZY* UNTUK TELUSUR TRAJEKTORI PADA UAV *QUADCOPTER*

Nama : Hanief Sasongko
Dosen Pembimbing 1 : Ir. Rusdhianto Effendie A.K, M.T.
Dosen Pembimbing 2 : Mochammad Sahal, S.T., M.Sc.

ABSTRAK

Quadcopter merupakan salah satu jenis *Unmanned Aerial Vehicle* (UAV) atau pesawat tanpa awak yang memiliki empat buah motor *propeller* sebagai penggeraknya. Permasalahan ketepatan dalam melakukan gerak lateral menelusuri trajektori yang diinginkan merupakan hal yang cukup penting dalam pengembangan *Quadcopter*.

Berdasarkan permasalahan tersebut, dalam tugas akhir ini dirancang kontroler *Neurofuzzy* untuk meminimalisir ketidaktepatan saat *Quadcopter* menelusuri trajektori secara otomatis. Desain kontroler *Neurofuzzy* pada tugas akhir ini menggunakan metode mamdani. Kontroler *Neurofuzzy* digunakan untuk telusur trajektori *Quadcopter* mengikuti sinyal referensi, sedangkan kontroler PD digunakan untuk mengatur kestabilan *Quadcopter*.

Hasil pengujian kontroler menunjukkan bahwa sistem hasil desain dapat mengatur gerak *Quadcopter* sesuai dengan lintasannya dengan nilai *error* lintasan pada sumbu x sebesar ± 0.01 m dengan nilai RMSE 0.625 dan *error* lintasan pada sumbu y sebesar ± 0.01 m dengan nilai RMSE 0.596. Sedangkan untuk pengujian dengan *noise* 25%, menghasilkan gerak *Quadcopter* dengan *error* lintasan pada sumbu x sebesar ± 0.05 m dengan nilai RMSE 0.638 dan *error* lintasan pada sumbu y sebesar ± 0.05 m dengan nilai RMSE 0.613.

Kata Kunci: *Quadcopter*, Telusur Trajektori, Metode Mamdani, Kontroler *Neurofuzzy*.

--halaman ini sengaja dikosongkan--

DESIGN OF NEUROFUZZY CONTROLLER FOR TRAJECTORY KEEPING OF THE UAV QUADCOPTER

Name : Hanief Sasongko
Supervisor I : Ir. Rusdhianto Effendie A.K, M.T.
Supervisor II : Mochammad Sahal, S.T., M.Sc.

ABSTRACT

Quadcopter is a type of Unmanned Aerial Vehicle (UAV), or drone which has four piece of propeller motors as their driving force. Problems accuracy in performing lateral movements searching the desired trajectory is quite important in the development of Quadcopter.

Based on these problems, in this thesis Neurofuzzy controller designed to minimize inaccuracies when Quadcopter automatically discover trajectory. Neurofuzzy controller design in this thesis using mamdani methods. Neurofuzzy controller is used to search the trajectory Quadcopter follow the reference signal, while the PD controller is used to regulate the stability of Quadcopter.

The test results show that the system controller can adjust the design results in accordance with the trajectory of motion Quadcopter with the track error value on the x axis ± 0.01 m with RMSE values 0.625 and a track error on the y axis ± 0.01 m with RMSE values 0.596. As for noise testing by 25%, resulting in motion Quadcopter with the track error on the x axis ± 0.05 m with RMSE values 0638 and error trajectory on the y axis ± 0.05 m with RMSE value 0613.

Keywords: *Quadcopter, Trajectory Keeping, Mamdani Method, Neurofuzzy Controller.*

--halaman ini sengaja dikosongkan--

KATA PENGANTAR

Segala puji kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya dalam usaha dan kerja keras sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul :

“Perancangan Kontroler *Neurofuzzy* Untuk Telusur Trajektori Pada UAV *Quadcopter*”

Tugas Akhir ini disusun guna memenuhi persyaratan untuk menyelesaikan studi Teknik Elektro di Bidang Studi Teknik Sistem Pengaturan, Program Sarjana Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam penyusunan laporan Tugas Akhir ini, penulis banyak mendapatkan bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis dengan tulus ikhlas menyampaikan banyak terima kasih kepada:

1. Bapak Ir. Rusdhianto Effendie A.K., M.T. sebagai dosen pembimbing 1 dan Bapak Mochammad Sahal, ST, M.Sc. sebagai dosen pembimbing 2 penulis, terima kasih atas segala kesabaran dan kesediaannya meluangkan waktu untuk membimbing serta memberi ilmu yang bermanfaat sehingga Tugas Akhir ini dapat terselesaikan.
2. Orang tua dan juga keluarga yang selalu mendukung sehingga Tugas Akhir ini dapat terselesaikan.
3. Dan semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah memberi dorongan dan bantuan dalam menyelesaikan Tugas Akhir ini baik secara langsung maupun tidak langsung.

Dengan segala kerendahan hati, kami berharap apa yang ada dalam buku Tugas Akhir ini dapat bermanfaat, dan berguna sebagai sumbangan pikiran bagi kita semua dalam berprestasi turut mengisi pembangunan Bangsa dan Negara.

Surabaya, Januari 2017

Penulis

--halaman ini sengaja dikosongkan--

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
ABSTRAK (ENGLISH)	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xxi
 BAB I PENDAHULUAN.....	 1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi	3
1.6 Sistematika Penulisan.....	3
1.7 Relevansi.....	4
 BAB II DASAR TEORI.....	 5
2.1 Tinjauan Pustaka	5
2.2 <i>Quadcopter</i>	5
2.2.1 Sejarah <i>Quadcopter</i>	6
2.2.2 Sistem perangkat keras <i>Quadcopter</i>	7
2.2.3 Sistem perangkat lunak <i>Quadcopter</i>	12
2.2.4 Konsep Dasar Pergerakan <i>Quadcopter</i>	12
2.3 Pemodelan <i>Quadcopter</i>	15
2.3.1 Kinematika <i>Quadcopter</i>	16
2.3.2 Dinamika <i>Quadcopter</i>	23
2.4 Logika <i>Fuzzy</i>	25
2.4.1 Himpunan <i>Fuzzy (Fuzzy Set)</i>	26
2.4.2 Fungsi Keanggotaan (<i>membership function</i>).....	26
2.4.3 Kontroler Logika <i>Fuzzy</i>	28
2.5 Jaringan Saraf Tiruan	31
2.6 Kontroler <i>Neurofuzzy</i>	32
2.6.1 Tahap Training.....	33
2.6.2 Tahap <i>Forward Propagation</i>	34

2.6.3	Tahap <i>Backward Propagation</i>	35
2.6	Kontroler PD	36
BAB III PERANCANGAN SISTEM		37
3.1	Desain Acuan <i>Quadcopter</i> Pada Simulasi.....	37
3.2	Identifikasi Konstanta	38
3.2.1	Pengukuran Kecepatan Motor dan <i>Propeller</i>	38
3.2.2	Pengukuran Gaya Angkat Motor dan <i>Propeller</i>	39
3.2.3	Pengukuran Gaya Thrust	40
3.2.4	Pengukuran Gaya Drag	41
3.2.5	Perhitungan Konstanta Inersia	42
3.3	Model Matematika Hasil Identifikasi.....	43
3.4	Perancangan Kontroler.....	45
3.5	Perancangan Kontroler PD Stabilisasi Gerak Rotasi.....	46
3.5.1	Perancangan Kontroler PD Stabilisasi sudut <i>Roll</i>	46
3.5.2	Perancangan Kontroler PD Stabilisasi sudut <i>Pitch</i>	47
3.5.3	Perancangan Kontroler PD Stabilisasi sudut <i>Yaw</i>	48
3.6	Perancangan Kontroler <i>Neurofuzzy</i>	48
3.6.1	Perancangan Kontroler <i>Neurofuzzy</i> untuk Sumbu <i>x</i>	49
3.6.2	Perancangan Kontroler <i>Neurofuzzy</i> untuk Sumbu <i>y</i>	51
3.6.3	Perancangan Kontroler Sumbu <i>z</i>	53
BAB IV HASIL PENGUJIAN DAN ANALISIS SISTEM		55
4.1	Simulasi Pergerakan <i>Quadcopter</i>	55
4.2	Simulasi Stabilisasi Gerak <i>Quadcopter</i>	56
4.3	Simulasi Telusur Trajektori Gerak <i>Quadcopter</i>	58
4.3.1	Telusur Trajektori Gerak Translasi <i>x</i> dan <i>y</i>	58
4.3.2	<i>Altitude</i> Sumbu <i>z</i>	60
4.3.3	Telusur Trajektori Gerak <i>Quadcopter</i> dengan <i>Noise</i>	61
BAB V PENUTUP		65
5.1	Kesimpulan	65
5.2	Saran	65
DAFTAR PUSTAKA		67
LAMPIRAN A		A1
A.1	Program <i>Simulink</i> Keseluruhan Sistem	A1
A.2	Program <i>Simulink</i> Pengendalian Sudut <i>Roll</i>	A1

A.3	Program <i>Simulink</i> Pengendalian Sudut <i>Pitch</i>	A2
A.4	Program <i>Simulink</i> Pengendalian Sudut <i>Yaw</i>	A2
A.5	Program <i>Simulink</i> Pengendalian Posisi <i>x</i>	A2
A.6	Program <i>Simulink</i> Pengendalian Posisi <i>y</i>	A3
A.7	Program <i>Simulink</i> Pengendalian Posisi <i>z</i>	A3
A.8	Program <i>Simulink Cascade Neurofuzzy</i> Sumbu <i>x</i> dan <i>y</i> ...	A3
A.9	Program <i>Simulink</i> Gerak Rotasi	A4
A.9	Program <i>Simulink</i> Gerak Translasi.....	A4
LAMPIRAN B		B1
B.1	Program <i>Neurofuzzy</i> pada MATLAB.....	B1
B.2	Program <i>Convert U to Omega</i> pada MATLAB	B3
RIWAYAT PENULIS		C1

--halaman ini sengaja dikosongkan--

DAFTAR GAMBAR

Gambar 2.1 <i>De Bothezat Quadrotor (1923)</i>	6
Gambar 2.2 <i>Bell Boeing TiltRotor (QTR)</i>	6
Gambar 2.3 Produk dari <i>3DRobotics Solo</i>	7
Gambar 2.4 Sistem Elektronik <i>Quadcopter</i>	8
Gambar 2.5 Frame Taloon	9
Gambar 2.6 APM 2.6	9
Gambar 2.7 ESC TBS Bulletproof	10
Gambar 2.8 Motor <i>Brushless Sunnysky 900kV</i>	10
Gambar 2.9 Modul Transmitter dan Receiver remote control	11
Gambar 2.10 Modul Telemetry	12
Gambar 2.11 <i>Ground Station Program Mission Planner</i>	12
Gambar 2.12 <i>Quadcopter</i> pada keadaan <i>hover</i>	13
Gambar 2.13 <i>Quadcopter</i> pada Gerak <i>pitch</i>	14
Gambar 2.14 <i>Quadcopter</i> pada Pergerakan <i>roll</i>	14
Gambar 2.15 <i>Quadcopter</i> pada Pergerakan <i>yaw</i>	15
Gambar 2.16 Konfigurasi <i>Quadcopter</i> b-frame terhadap e-frame... ..	16
Gambar 2.17 <i>Earth-frame</i> dan <i>Body-frame</i> dari <i>Quadcopter</i>	17
Gambar 2.18 Rotasi <i>Quadcopter</i> Sumbu <i>z</i>	18
Gambar 2.19 Rotasi <i>Quadcopter</i> Sumbu <i>y</i>	19
Gambar 2.20 Rotasi <i>Quadcopter</i> Sumbu <i>x</i>	20
Gambar 2.21 Kontrol Logika Fuzzy	28
Gambar 2.22 Contoh Fuzzifikasi dengan 5 himpunan	29
Gambar 2.23 Struktur Model Jaringan Saraf Tiruan	31
Gambar 2.24 Struktur Formulasi <i>Forward Propagation</i>	31
Gambar 2.25 Struktur Formulasi <i>Backward Propagation</i>	32
Gambar 2.26 Struktur Kontroler <i>Neurofuzzy</i> metode mamdani	33
Gambar 3.1 Desain acuan <i>Quadcopter</i>	37
Gambar 3.2 Hubungan kecepatan motor dan PWM	39
Gambar 3.3 Hubungan kecepatan motor dan Gaya Angkat	40
Gambar 3.4 Massa, Tinggi, dan Jari-jari <i>Quadcopter</i>	42
Gambar 3.5 Diagram blok sistem keseluruhan	45
Gambar 3.6 Diagram blok kontrol sudut <i>roll</i>	46
Gambar 3.7 Diagram blok kontrol sudut <i>pitch</i>	47
Gambar 3.8 Diagram blok kontrol sudut <i>yaw</i>	48
Gambar 3.9 Struktur <i>Neurofuzzy</i> Mamdani	49
Gambar 3.10 Diagram blok kontroler <i>Neurofuzzy</i> pada sumbu <i>x</i>	49
Gambar 3.11 Dua variabel dengan 3 <i>membership function</i>	50

Gambar 3.12	Diagram blok kontroler <i>Neurofuzzy</i> pada sumbu y	51
Gambar 3.13	Dua variabel dengan 3 <i>membership function</i>	52
Gambar 3.14	Diagram blok kontrol gerak translasi pada sumbu z ..	53
Gambar 4.1	Simulasi perancangan sistem telusur trajektori <i>Quadcopter</i>	55
Gambar 4.2	Pengujian stabilisasi gerak <i>roll</i>	56
Gambar 4.3	Pengujian stabilisasi gerak <i>pitch</i>	57
Gambar 4.4	Pengujian stabilisasi gerak <i>yaw</i>	56
Gambar 4.5	Grafik sinyal trajektori gerak translasi pada sumbu x ..	58
Gambar 4.6	Grafik sinyal trajektori gerak translasi pada sumbu y ..	58
Gambar 4.7	Hasil telusur trajektori <i>Quadcopter</i> tanpa <i>noise</i>	60
Gambar 4.8	Grafik sinyal posisi sumbu z	61
Gambar 4.9	Grafik sinyal trajektori gerak translasi pada sumbu x dengan <i>noise</i>	62
Gambar 4.10	Grafik sinyal trajektori gerak translasi pada sumbu y dengan <i>noise</i>	63
Gambar 4.11	Hasil telusur trajektori <i>Quadcopter</i> dengan <i>noise</i>	64

DAFTAR TABEL

Tabel 3.1 Hasil pengukuran kecepatan motor <i>propeller</i>	38
Tabel 3.2 Hasil Pengukuran Gaya angkat motor <i>propeller</i>	39
Tabel 3.3 Identifikasi konstanta	44
Tabel 3.4 Nilai K_p dan τ_d kontroler PD sudut <i>roll</i>	47
Tabel 3.5 Nilai K_p dan τ_d kontroler PD sudut <i>pitch</i>	48
Tabel 3.6 Nilai K_p dan τ_d kontroler PD sudut <i>yaw</i>	48
Tabel 4.1 Nilai K_p dan τ_d Kontroler PD Stabilisasi <i>Quadcopter</i>	56
Tabel 4.2 Hasil <i>error</i> RMSE	59
Tabel 4.3 Nilai K_p dan τ_d kontroler PD untuk sumbu <i>z</i>	60
Tabel 4.4 Nilai gangguan yang diberikan.....	62
Tabel 4.5 Hasil simulasi gerak <i>Quadcopter</i> dengan varian <i>noise</i>	63

--halaman ini sengaja dikosongkan--

BAB I

PENDAHULUAN

1.1 Latar Belakang

Unmanned aerial vehicle (UAV) atau disebut juga sebagai pesawat terbang tanpa awak akhir-akhir ini menjadi topik yang sangat menarik untuk dibahas. Aplikasi penggunaan UAV dalam dunia *aeromodeling* saat ini dapat difungsikan seperti untuk misi militer, pengawasan dan pengintaian, pemetaan suatu wilayah, fotografi dan video udara, dan masih banyak lagi fungsi lainnya. Pada setiap penggunaannya, UAV pasti memiliki sistem kontrol dan tingkat *maneuverability* yang berbeda-beda pada tiap jenisnya. Pada perkembangannya, *Quadcopter* dirancang untuk kendaraan udara tanpa awak yang dapat dikendalikan jarak jauh oleh atau tanpa seorang pilot (*autopilot*). *Quadcopter* merupakan pesawat yang memiliki empat buah motor yang dikendalikan secara terpisah satu sama lainnya.

Quadcopter sebagaimana fungsinya harus memiliki keseimbangan yang baik saat terbang, terutama pada gerak rotasi dan translasi yang sangat mempengaruhi terbang *Quadcopter*. *Quadcopter* merupakan jenis helikopter dengan empat buah motor yang dipasang simetris pada ujung-ujung kerangka utama. Motor depan dan belakang berputar searah jarum jam (*clockwise*), sedangkan motor kanan dan motor kiri berputar berlawanan arah jarum jam (*counter clockwise*). Seiring dengan perkembangannya, saat ini sudah banyak dilakukan riset mengenai konfigurasi *Quadcopter* dengan berbagai macam *frame* pendukungnya. Beberapa konfigurasi *frame Quadcopter* tersebut yaitu *frame* konfigurasi “*plus(+)*”, konfigurasi “*X*”, dan konfigurasi “*H*”. dengan berbedanya struktur *frame* tersebut, berbeda pula cara kontrol pada masing-masing konfigurasi *frame* tersebut.

Secara garis besar ada tiga macam kategori yang harus diselesaikan dalam pengembangan UAV, yaitu antara lain efisiensi aerodinamika, peningkatan pembebanan, masalah kontrol dan stabilitas. Ada dua macam gerak terbang pada *Quadcopter*, yaitu gerakan longitudinal dan gerak lateral. Pergerakan longitudinal terbagi dalam tiga fase utama, yaitu *take-off* (tingal landas), *hovering* (melayang) dan *landing* (pendaratan). Dari kedua gerakan terbang tersebut, gerakan lateral merupakan fase yang cukup kritis dan masih perlu dilakukan banyak pengembangan.

Gerak translasi pada *Quadcopter* di tentukan oleh resultan gaya dan sudut-sudut pada *Quadcopter*. Jika dilakukan pengendalian gerak translasi, maka dari pernyataan sebelumnya dapat diambil hipotesa bahwa sinyal kontrol harus bergantung pada evaluasi gerak translasinya dan besar sudut-sudut *Quadcopter*. Terdapat beberapa metode untuk keadaan seperti ini, dalam hal ini digunakan kontroler *Neurofuzzy*.

1.2 Permasalahan

Pergerakan yang lincah dapat ditandai dengan kemampuan *Quadcopter* untuk bergerak secara lateral maupun longitudinal. Pergerakan tersebut dapat dicapai dengan menggunakan metode kontrol yang akan diterapkan pada *Quadcopter*. Mengatasi ketidaktepatan *Quadcopter* saat melakukan telusur trajektori merupakan suatu permasalahan yang diangkat pada buku tugas akhir ini. Penggunaan metode kontrol *Neurofuzzy* pada penelitian ini digunakan untuk mengatasi permasalahan tersebut.

1.3 Batasan masalah

Parameter kontrol pada gerak *Quadcopter* sangat banyak, untuk itu diperlukan batasan untuk mencari solusi yang tepat pada masalah yang diangkat. Batasan masalahnya antara lain yaitu :

- a. Perancangan kontroler dilakukan melalui tahap simulasi terlebih dahulu dengan bantuan *software*.
- b. Pergerakan *Quadcopter* untuk telusur lintasan telah direncanakan sebelumnya.
- c. Konfigurasi *frame Quadcopter* yang digunakan berbentuk “*plus(+)*”
- d. Gerak yang dikontrol yaitu gerak *roll*, *pitch* untuk pergerakan maju, mundur serta menyamping. Gerak *yaw* akan dijaga konstan karena pada penelitian ini pergerakan *heading Quadcopter* tidak termasuk dalam perancangan kontroler.

1.4 Tujuan

Pergerakan *autonomous Quadcopter* yang tepat sesuai dengan trajektori sangat diperlukan saat melakukan manuver. Tujuan pada penelitian ini yaitu mengatasi ketidaktepatan *Quadcopter* saat melakukan telusur trajektori (membuat *Quadcopter* bergerak sesuai dengan lintasan yang telah ditentukan dengan tepat) menggunakan kontroler *Neurofuzzy*.

1.5 Metodologi

Metodologi yang digunakan pada penelitian ini yaitu sebagai berikut:

- a. Studi literatur
Studi literatur digunakan untuk mencari informasi atau data mengenai *plant*, kontroler atau sistem secara keseluruhan. Hal ini dapat dilakukan dengan melihat acuan dari buku teks, jurnal, internet, dan lain-lain.
- b. Pemodelan dan identifikasi sistem
Model matematis dari *plant* diperoleh dari pemodelan fisis analisis kinematik dan dinamika dari *Quadcopter*, dan konstanta yang terdapat dalam persamaan diperoleh dari identifikasi *plant* atau informasi dari *datasheet* pada tiap-tiap komponennya.
- c. Desain kontroler dan simulasi
Dari model matematik yang diperoleh maka akan diperoleh parameter-parameter *plant* sehingga dapat disimulasikan untuk mendapatkan responnya setelah menggunakan kontroler.
- d. Pengujian dan analisa
Tahapan ini dilakukan untuk menguji sistem yang telah dibuat. Data yang didapat dari pengujian kemudian dianalisa.
- e. Penulisan Buku
Pada tahap ini dilakukan penyusunan buku sesuai dengan data-data yang sudah didapatkan dan analisa-analisa dari pengujian.

1.6 Sistematika penulisan

Penulisan tugas akhir ini disusun dalam lima bab yang masing-masing membahas permasalahan yang berhubungan dengan tugas akhir yang telah dibuat dengan sistematika penulisan sebagai berikut:

- a. BAB 1 PENDAHULUAN
Berisi tentang latar belakang, permasalahan, batasan masalah, tujuan, metodologi, sistematika penulisan dan relevansi pembahasan tugas akhir ini.
- b. BAB 2 DASAR TEORI
Memberi gambaran secara umum mengenai konsep teori yang mendasari perancangan tugas akhir ini, meliputi teori definisi *Quadcopter*, komponen elektronik *Quadcopter*, dan teori sistem kontrol *Neurofuzzy*.
- c. BAB 3 PERANCANGAN SISTEM
Dijelaskan secara detail tentang spesifikasi sistem, identifikasi kebutuhan, desain *Quadcopter*, pemodelan sistem, perancangan

simulasi *Quadcopter*, perancangan desain sistem kontrol *Neurofuzzy* pada *Quadcopter*.

d. **BAB 4 PENGUJIAN DAN ANALISA SISTEM**

Berisi prosedur pelaksanaan pengujian dan analisis data dari proses identifikasi sampai dengan tahap pengujian kontroler *Neurofuzzy*.

e. **BAB 5 PENUTUP**

Berisi tentang kesimpulan dan saran-saran untuk pengembangan tugas akhir ini kedepannya.

1.7 Relevansi

Hasil yang diperoleh dari tugas akhir ini diharapkan menjadi referensi pengembangan teknologi navigasi *aeromodeling* UAV terutama pada *Quadcopter*, dan juga metode kontrol *Neurofuzzy* ini dapat menjadi pembanding penggunaan metode kontrol pada *Quadcopter* lainnya.

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang berkaitan dengan topik penelitian *Quadcopter*. Topik yang dibahas meliputi sejarah *Quadcopter*, pergerakan *Quadcopter*, perangkat keras penyusun *Quadcopter*, dan kontroler yang digunakan pada *Quadcopter*.

2.1. Tinjauan Pustaka

Quadcopter merupakan jenis UAV yang memiliki empat buah motor sebagai penggerakannya.[8] *Quadcopter* dapat bergerak secara otomatis maupun manual dengan bantuan *remote control*. Banyak penelitian yang telah dilakukan untuk mengembangkan navigasi gerak lateral otomatis dari *Quadcopter*. Seperti pada penelitian sebelumnya yang dilakukan oleh Wira Tama dalam tugas akhirnya, telah dibahas mengenai navigasi *waypoint Quadcopter* dengan menggunakan kontroler PID.[3] Dalam hasil penelitian tersebut menghasilkan pergerakan *waypoint Quadcopter* mampu dilakukan dengan kontroler PID, namun masih terdapat *error* pada lintasan. Dalam perancangan tugas akhir kali ini akan dibuat perencanaan navigasi trajektori lintasan dari titik awal menuju ke titik-titik yang telah ditentukan dengan menggunakan kontroler *Neurofuzzy*.

Perancangan lintasan *Quadcopter* ini dilakukan dengan cara menyusun algoritma program dan kemudian menentukan posisi awal pergerakan *Quadcopter*. Penggunaan kontroler *Neurofuzzy* digunakan untuk membuat *Quadcopter* berada pada jalur lintasan yang telah dibuat dengan tepat.

2.2. *Quadcopter*

Quadcopter adalah pesawat tanpa awak yang menggunakan empat motor sebagai penggerak baling-baling (*propeller*). *Quadcopter* memiliki kemampuan yang tidak dimiliki oleh UAV yang memiliki sayap (*wing*) atau pesawat *fixed wings*, yaitu dapat melakukan *take off* dan *landing* secara vertikal.[4] *Vertical Take Off Landing (VTOL) Aircraft* merupakan pemberian nama untuk jenis pesawat yang dapat melakukan *take off* dan *landing* tegak lurus terhadap bumi. Dengan adanya hal tersebut memungkinkan *Quadcopter* dapat terbang pada tempat yang sempit, sehingga pergerakannya dapat menjadi sangat lincah.

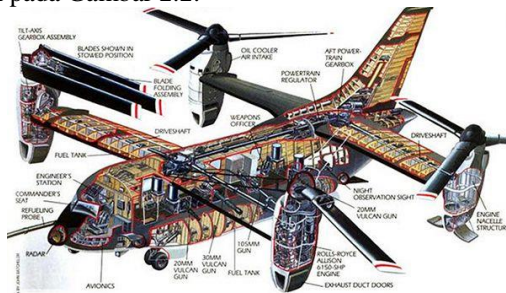
2.2.1. Sejarah *Quadcopter*

Riset mengenai *Quadcopter* awalnya dibuat pada tahun 1923 oleh George DeBothezat. *Quadcopter* tersebut diberi nama *De Bothezat Quadrotor* yang bentuknya ditunjukkan pada Gambar 2.1.



Gambar 2.1 *De Bothezat Quadrotor (1923)*

Perkembangan *Quadcopter* dari tahun ke tahun sangat pesat, terbukti perusahaan yang bergerak di bidang penerbangan, *Bell Helicopter* dan *Boeing* menciptakan *Bell Boeing Quad TiltRotor (QTR)*. *Bell Boeing Quad TiltRotor* adalah pesawat dengan penggerak empat buah motor sebagai penggerak utamanya. [4] Pesawat tersebut dapat menampung kapasitas setara dengan pesawat C-130 Hercules, dan dapat terbang pada kecepatan 250 knot. Model dari *Bell Boeing Quad TiltRotor* ini ditunjukkan pada Gambar 2.2.



Gambar 2.2 *Bell Boeing Quad TiltRotor (QTR)*

Pada saat ini *Quadcopter* telah banyak diperbincangkan masyarakat umum. *Quadcopter* berukuran kecil atau mini yang dikendalikan melalui *remote control* banyak digunakan masyarakat untuk keperluan seperti fotografi, pemetaan, atau sekedar hobi. Salah satu perusahaan yang memproduksi *Quadcopter* mini adalah *3DRobotics*. *Quadcopter* mini

yang dijual dipasaran saat ini sangat banyak macam dan fungsi kegunaannya pun juga berbeda-beda. Misalnya untuk keperluan fotografi, dimana *Quadcopter* diharuskan memiliki kamera dengan spesifikasi yang baik dan ditunjang dengan dudukan atau gimbal pada kamera yang dapat berputar sampai dengan 360°.

Quadcopter berukuran kecil tersebut diatur oleh sebuah *chip* mikrokontroler. Kecepatan kerja dari mikrokontroler yang biasanya dipakai mulai dari 8 Mhz hingga kecepatan mendekati kecepatan komputer yaitu 400 Mhz. Semakin cepat kecepatan kerja dari mikrokontroler yang digunakan, maka proses pengambilan data sensor, penghitungan keluaran kecepatan motor, keluaran sinyal motor, dan waktu pengiriman data ke *ground station* semakin cepat. Salah satu contoh dari *mini Quadcopter* dapat dilihat pada gambar 2.3.



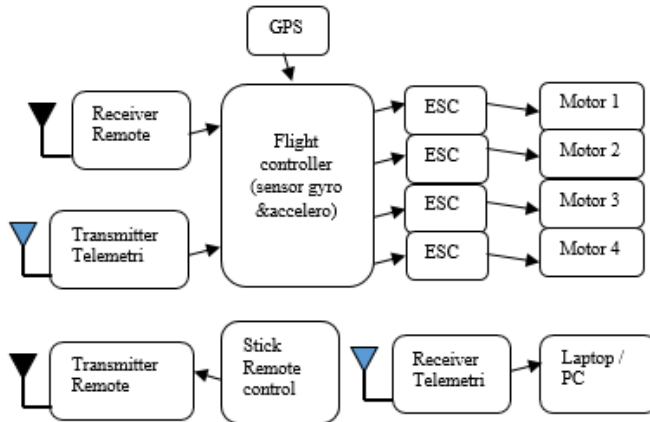
Gambar 2.3 Produk dari 3DRobotics Solo

2.2.2. Sistem Perangkat Keras *Quadcopter*

Sistem elektronik dan mekanik tidak pernah lepas dari sistem robot. Kedua sistem tersebut diintegrasikan agar robot dapat mencapai pergerakan yang diinginkan. Sama halnya dengan *Quadcopter*, komponen elektrik dan mekanik penyusun *Quadcopter* harus saling berintegrasi agar didapatkan pergerakan yang stabil dan lincah. *Quadcopter* dalam penelitian ini terdiri atas sistem kontroler berupa *flight controller*, telemetri sistem, motor *brushless*, driver motor *brushless* (ESC), dan *remote control* yang digunakan sebagai acuan dalam menentukan gerak terbangnya. Sensor yang digunakan adalah sensor sudut yang diukur menggunakan sensor *accelerometer*, dan terakhir sensor kecepatan sudut yang diukur menggunakan sensor *gyroscope*. Kedua sensor tersebut telah terintegrasi di dalam modul *flight controller*.

Rangkaian elektronika yang dirancang harus mampu menampung jumlah *input* dan *output* sensor serta komponen – komponen yang digunakan sebagai penunjang *Quadcopter*. Beberapa komponen seperti

sensor-sensor yang terdapat pada *flight controller*, *ESC* (*Electronic Speed Controller*), dan motor *brushless* merupakan modul yang dapat langsung digunakan dan hanya memerlukan kalibrasi pada tiap komponen tersebut. Perancangan keseluruhan dari sistem elektronika dari *Quadcopter* ditunjukkan pada Gambar 2.4.



Gambar 2.4 Sistem elektronik *Quadcopter*

Flight controller Board yang dipakai pada penelitian ini adalah APM Versi 2.6 yang dirancang oleh 3D Robotics. *Flight controller* ini sudah dilengkapi beberapa sensor dan *port* yang memudahkan untuk penggunaan sebagai komunikasi data. *Flight controller Board* APM Versi 2.6 dilengkapi dengan mikroprosesor ATmega 2560 sebagai pengolah data dari *flight controller* tersebut. Berikut merupakan komponen-komponen penyusun *Quadcopter*, yaitu :

a. Kerangka *Quadcopter* (*frame*)

Sistem mekanik yang *rigid* akan mendukung pergerakan *Quadcopter* menjadi lebih baik, oleh karena itu perancangan mekanik dalam hal ini *frame* dan dudukan motor haruslah proporsional dengan titik beban *Quadcopter*. Pada penelitian ini digunakan *frame* berjenis taloon dari TBS technology.



Gambar 2.5 *Frame Taloon*

b. *Flight controller APM* versi 2.6

APM menggunakan mikrokontroler Ardupilot Mega, dimana didalamnya sudah tertanam berbagai komponen dan modul sensor yang diperlukan untuk menerbangkan pesawat UAV. Selain itu, dalam APM ini sudah berisi program dasar dan *library Quadcopter* yang nantinya dapat diperbarui sistemnya melalui *flash firmware* pada *software APM*. *Software* bawaan APM ini yaitu *Mission Planner* atau *APM Planner*.



Gambar 2.6 *APM 2.6*

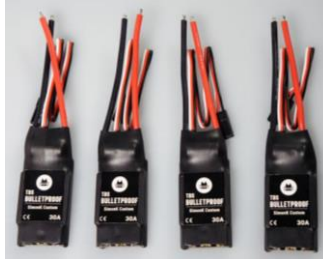
c. *Electronic speed controller (ESC)*

ESC atau disebut juga *Electronic Speed Control* adalah *driver* penggerak untuk jenis motor *brushless*, motor jenis ini biasanya digunakan pada bidang *aeromodeling* atau RC karena memiliki torsi yang cukup besar dibandingkan menggunakan motor DC biasa. Untuk melakukan *interface* dengan ESC, caranya cukup mudah, yaitu dengan memberikan input pulsa pada pin input sinyal ESC yang akan berpengaruh pada kecepatan motor *brushless*. Sinyal input yang diberikan ke ESC yaitu sinyal PWM yang dikendalikan oleh mikrokontroler. Sinyal PWM tersebut akan menggerakkan motor dengan kecepatan yang linier dengan besar pulsa yang diberikan.

Jenis ESC yang digunakan pada penelitian ini adalah *TBS Bulletproof 30A* yang memiliki spesifikasi maksimal arus sebesar 30 Ampere, seperti pada Gambar 2.7. Spesifikasi ESC *TBS Bulletproof 30A* adalah sebagai berikut :

Arus maksimal : 30 Ampere

Tegangan baterai : 6,4 – 16,8 VDC (li-po 3S-4S)



Gambar 2.7 ESC TBS Bulletproof

d. Motor *Brushless* DC

Motor *brushless* DC merupakan perangkat yang digunakan untuk memutar baling – baling atau *propeller Quadcopter*. Jenis motor ini menggunakan sumber listrik DC sebagai sumber energi. Pada prinsip kerjanya, motor *brushless* ini merupakan motor AC tiga fasa, di mana putaran pada motor disebabkan oleh medan magnet pada stator yang pada setiap saatnya hanya aktif dua fasa (hanya dua fasa yang ter-*supply* pada setiap saat sementara satu fasa lainnya tidak ter-*supply*). Motor *brushless* yang digunakan sebagai aktuatur pada penelitian tugas akhir ini adalah motor *brushless* DC Sunny Sky V2216 900 kV dan menggunakan *propeller* berukuran 10 cm x 4,5 cm. Bentuk motor *brushless* DC dan *propeller* terlihat seperti pada Gambar 2.8.



Gambar 2.8 Motor *Brushless* Sunnysky 900kV

- e. Modul *receiver* dan *transmitter* remote control
 Modul *Transmitter* dan *receiver* digunakan untuk operasi manual melakukan manuver pada *Quadcopter*. Hal ini dilakukan menggunakan *Remote Control (Analog)* yang sudah di kalibrasi sebelumnya sesuai dengan desain *Quadcopter* yang telah di buat. *Transmitter* berada pada sisi pengguna untuk memberikan nilai input kepada *Quadcopter*, sinyal tersebut akan diterima *receiver* dan diteruskan ke mikrokontroler berupa pulsa. Nantinya, mikrokontroler akan memberikan perintah ke driver motor untuk memutar motor melakukan gerak elektrik sesuai perintah. Tampilan *transmitter* dan *receiver* radio seperti pada Gambar 2.9.

Spesifikasi dari radio *transmitter* adalah:

Merek Remote Control	: Turnigy 9xR
Modul <i>Transmitter</i>	: DJT Frsky
<i>Channel</i>	: 9
Modulasi	: 2,4 GHz,
Baterai	: 2500 mAh Li-Po – 11,1 V

Sedangkan spesifikasi untuk *receiver* adalah :

Modul <i>receiver</i>	: Frsky V8FR
Ukuran modul utama	: 45mm x 26mm x 17mm,
Ukuran modul satelit	: 17mm x 23mm x 5mm,
Berat	: 15 gram,
<i>Range</i> tegangan	: 3,2 - 9,6 V,
Arus	: 20 mA,



Gambar 2.9 Modul *transmitter* dan *receiver* remote control

- f. Modul telemetri
 Modul telemetri ini berfungsi sebagai penghubung dan modul komunikasi data antara *Quadcopter* dan *ground station*. Modul ini sangat dibutuhkan apabila pengguna membutuhkan informasi

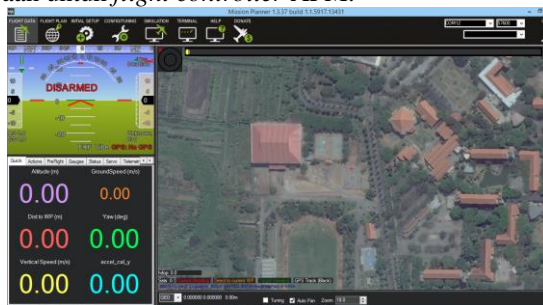
data dari *Quadcopter* saat terbang. Data-data sensor, kecepatan *Quadcopter*, jarak *Quadcopter* terhadap bumi pun dapat kita terima. Modul telemetri ini mempunyai dua modul, modul *transmitter* dan juga modul *receiver*. Modul *transmitter* dipasang pada *Quadcopter*, sedangkan modul *receiver* dipasang pada PC/*ground station*.



Gambar 2.10 Modul telemetri

2.2.3. Sistem Perangkat Lunak *Quadcopter*

Perangkat lunak sangat dibutuhkan untuk konfigurasi sebuah sistem mikrokontroler. Pemrograman, konfigurasi komponen, kalibrasi komponen, dan informasi data sensor semuanya dapat dilakukan dan dilihat pada perangkat lunak. Perangkat lunak yang digunakan pada penelitian ini yaitu Mission Planner, dimana perangkat ini merupakan program bawaan untuk *flight controller* APM.



Gambar 2.11 *Ground station* program Mission Planner

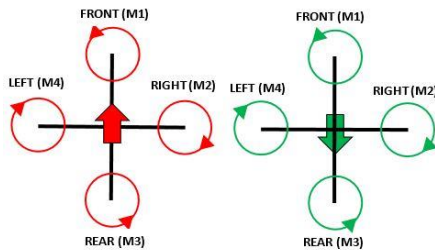
2.2.4. Konsep Dasar Pergerakan *Quadcopter*

Quadcopter merupakan salah satu jenis UAV yang memiliki gaya angkat dari empat buah motor yang menyusunnya. Untuk bergerak naik, diperlukan kecepatan yang sama dan torsi cukup besar pada keempat motornya. Secara garis besar, pergerakan *Quadcopter* terbagi menjadi empat, yaitu gerak *roll*, gerak *pitch*, gerak *yaw*, dan gerak *thrust*. Pada

penelitian ini digunakan *Quadcopter* konfigurasi *plus* “+”. Pada penjelasan pergerakan *Quadcopter* ini memiliki keterangan bahwa motor yang digambarkan berwarna hijau mempunyai kecepatan dibawah kecepatan nominal, dan untuk motor berwarna merah mempunyai kecepatan diatas kecepatan nominal, hal tersebut dapat dijelaskan sebagai berikut :

a. Gerak naik dan turun (Gerak *Thrust*)

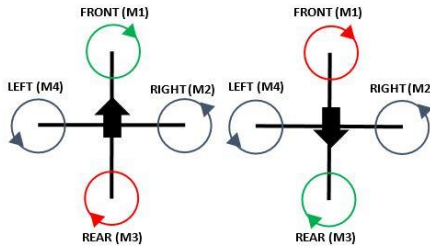
Pergerakan ini dilakukan dengan cara memberikan kecepatan yang sama pada masing-masing motor 1 s/d 4. Ketika kecepatan motor 1 s/d 4 melebihi kecepatan nominal (berwarna merah), maka *Quadcopter* akan bergerak naik, dan sebaliknya ketika kecepatan motor 1 s/d 4 lebih kecil dari kecepatan nominal (berwarna hijau) maka ketinggian *Quadcopter* akan menurun. Pergerakan *Quadcopter* pada keadaan *hover* ditunjukkan pada Gambar 2.12.



Gambar 2.12 *Quadcopter* pada keadaan *hover*

b. Gerak maju dan mundur (Gerak *Pitch*)

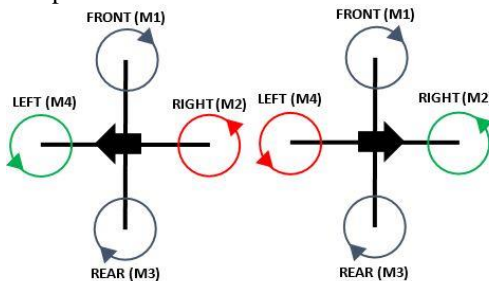
Pergerakan ini dilakukan dengan cara memberikan kecepatan yang berbeda pada motor 1 dan 3 (dengan menambah atau mengurangi kecepatan tiap motor). Sedangkan untuk motor 4 dan 2 memiliki kecepatan yang sama. Agar *Quadcopter* bergerak maju, maka motor 3 harus memiliki kecepatan lebih tinggi dari motor 1. Begitu pula sebaliknya jika *Quadcopter* bergerak mundur, maka motor 1 harus memiliki kecepatan lebih tinggi dari motor 3. Pergerakan *pitch Quadcopter* ditunjukkan pada Gambar 2.13.



Gambar 2.13 *Quadcopter* pada gerak *pitch*

c. Gerak kekanan dan kekiri (Gerak *Roll*)

Pergerakan ini dilakukan dengan cara memberikan kecepatan yang berbeda pada motor 2 dan 4 (dengan menambah atau mengurangi kecepatan tiap motor tersebut). Sedangkan untuk motor 1 dan 3 memiliki kecepatan yang sama. Agar *Quadcopter* bergerak ke kiri, maka motor 2 harus memiliki kecepatan lebih tinggi dari motor 4. Begitu pula sebaliknya jika *Quadcopter* bergerak ke kanan, maka motor 4 harus memiliki kecepatan lebih tinggi dari motor 2. Pergerakan *roll Quadcopter* ditunjukkan pada Gambar 2.14.

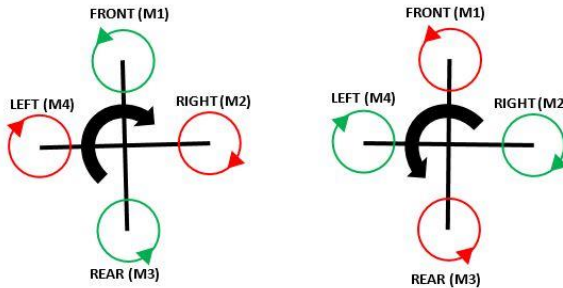


Gambar 2.14 *Quadcopter* pada pergerakan *roll*

d. Gerak berputar searah jarum jam dan berlawanan arah jarum jam (Gerak *yaw*)

Pergerakan ini dilakukan dengan cara memberikan kecepatan yang berbeda pada tiap motor (dengan menambah atau mengurangi kecepatan tiap motor). Agar *Quadcopter* bergerak berputar ke kiri, maka motor 1 dan motor 3 harus memiliki kecepatan lebih tinggi dari motor 2 dan motor 4. Begitu pula

sebaliknya jika *Quadcopter* berputar ke kanan, maka motor 2 dan motor 4 harus memiliki kecepatan lebih tinggi dari motor 1 dan motor 3. Pergerakan yaw *Quadcopter* ditunjukkan pada Gambar 2.15.



Gambar 2.15 *Quadcopter* pada pergerakan yaw

2.3. Pemodelan *Quadcopter* [1]

Sebagaimana telah diketahui bahwa *Quadcopter* merupakan helikopter dengan empat buah motor *propeller*. Motor *propeller* tersebut digunakan untuk mengangkat *Quadcopter* agar bisa terbang dan motor tersebut ditempatkan pada keempat buah sisi *Quadcopter* dengan jarak dari titik tengah adalah sama. Untuk dapat melakukan pemodelan dari *Quadcopter* dapat dilakukan dengan cara mengasumsikan beberapa parameter yaitu:

- Struktur dari *Quadcopter* dianggap *rigid*/kaku
- Struktur dari *Quadcopter* dianggap simetris
- Struktur motor dan *center body* dianggap lingkaran
- Titik berat beban *Quadcopter* diasumsikan berada tepat di tengah pusat massa *Quadcopter*.
- Efek getaran masing-masing *propeller* dianggap tidak terjadi
- Efek aerodinamis dianggap tidak mempengaruhi *Quadcopter*.

Quadcopter memiliki 6 *degree of freedom* (DOF) dengan 12 keluaran, 6 keluaran dari 12 keluaran ini menentukan pergerakan dari *Quadcopter*. Variabel-variabel tersebut adalah:

- x = posisi *Quadcopter* terhadap sumbu x_e
- y = posisi *Quadcopter* terhadap sumbu y_e
- z = posisi *Quadcopter* terhadap sumbu z_e
- u = kecepatan *Quadcopter* yang diukur pada sumbu x_b
- v = kecepatan *Quadcopter* yang diukur pada sumbu y_b

w = kecepatan *Quadcopteryang* diukur pada sumbu z_b

ϕ = sudut *roll* terhadap sumbu x_e

θ = sudut *pitch* terhadap sumbu y_e

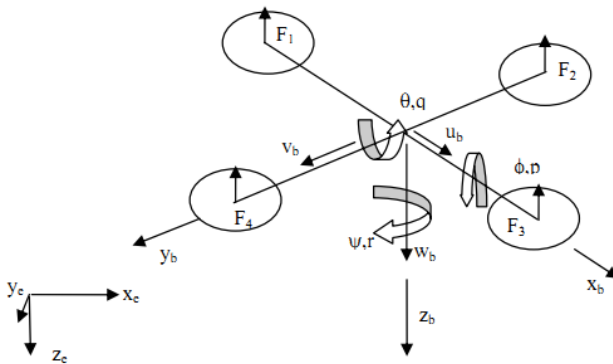
ψ = sudut *yaw* terhadap sumbu z_e

p = kecepatan sudut *roll* yang diukur pada sumbu x_b

q = kecepatan sudut *pitch* yang diukur pada sumbu y_b

r = kecepatan sudut *yaw* yang diukur pada sumbu z_b

Guna menentukan acuan-acuan dinamika pada *Quadcopter* perlu adanya dua sistem koordinat yang dibutuhkan untuk mendefinisikan keadaan dari *Quadcopter*. Pertama, sumbu x_b , y_b , dan z_b yang terletak pada titik gravitasi (*COG*) *Quadcopter*. Kedua, sistem koordinat yang tetap yaitu pusat bumi, yaitu sumbu x_e , y_e , dan z_e). Jadi ada 2 frame, yaitu *body frame* (*b-frame*, *frame* badan *Quadcopter*) dan *earth frame* (*e-frame*, *frame* bumi). Acuan ini ditunjukkan pada Gambar 2.16.



Gambar 2.16 Konfigurasi *Quadcopter b-frame* terhadap *e-frame*

2.3.1. Kinematika *Quadcopter*[1]

Pada analisis kinematika ini dilakukan beberapa penetapan, yaitu analisis pada penelitian ini menggunakan bingkai diagram cartesius dengan tiga dimensi (x, y, z). Bingkai diagram cartesius tersebut dibagi menjadi dua, yaitu bingkai bumi (*earth frame*) yang tidak bergerak, serta bingkai *body Quadcopter* yang bergerak rotasi dan translasi. Kemudian menetapkan sumbu x -bumi berada pada garis utara-selatan, sumbu y -bumi berada pada garis timur-barat, dan sumbu z -bumi berada pada garis vertikal menuju pusat bumi. Sedangkan untuk sumbu x -*Quadcopter* berada pada garis depan-belakang *Quadcopter*, sumbu y -*Quadcopter*

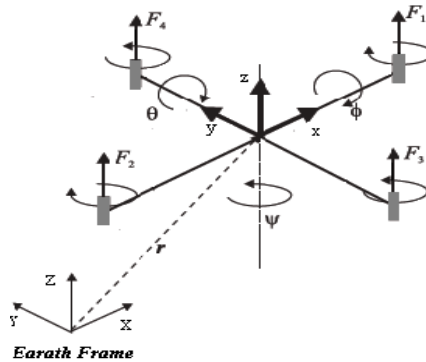
berada pada garis kiri-kanan *Quadcopter*, dan sumbu z -*Quadcopter* berada pada garis atas-bawah *Quadcopter*.

Jika pendefinisian sumbu x^+ menuju utara dan depan *Quadcopter*, sumbu z^+ menjauhi pusat bumi dan menuju atas *Quadcopter*, maka berdasarkan aturan kaidah tangan kanan *cross vector* dapat ditentukan bahwa sumbu y^+ menuju barat atau kiri *Quadcopter* seperti yang ditampilkan pada Gambar 3.3.

Posisi linear *Quadcopter* (Γ^e) ditentukan dari koordinat vektor antara origin *B-frame* serta origin dari *E-frame* dengan memperhatikan *E-frame*. Posisi angular *Quadcopter* (Θ^e) ditentukan dari orientasi *b-frame* terhadap *e-frame*. Posisi linear dan posisi angular terdapat pada Persamaan 2.1-2.2.

$$\Gamma^E = [x \ y \ z]^T \quad (2.1)$$

$$\Theta^E = [\phi \ \theta \ \psi]^T \quad (2.2)$$

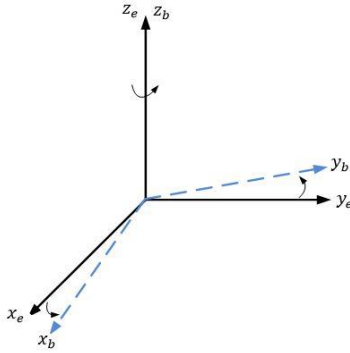


Gambar 2.17 *Earth-frame* dan *Body-frame* dari *Quadcopter*

Matriks rotasi dari *Quadcopter* diperoleh dengan mengkalikan matriks transformasi untuk rotasi di tiap sumbu.

a. Rotasi Sumbu z

Quadcopter berotasi pada sumbu z dan menghasilkan sudut *yaw* yang dilambangkan dengan $R(\psi, z)$. Rotasi sudut *yaw* terdapat pada Gambar 2.18



Gambar 2.18 Rotasi *Quadcopter* Sumbu z

$$\begin{cases} x_e = x_b \cos \psi - y_b \sin \psi \\ y_e = x_b \sin \psi + y_b \cos \psi \\ z_e = z_b \end{cases} \quad (2.3)$$

Persamaan 2.3 dapat dibuat dalam bentuk matriks pada Persamaan 2.4.

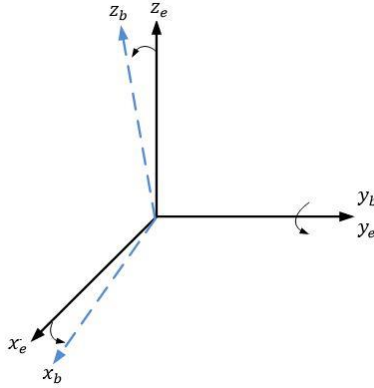
$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \quad (2.4)$$

Sehingga didapatkan matriks rotasi pada sumbu x ditunjukkan pada Persamaan 2.5.

$$R(\psi, z) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

b. Rotasi Sumbu y

Quadcopter berotasi pada sumbu y dan menghasilkan sudut *pitch* yang dilambangkan dengan $R(\theta, y)$. Rotasi sudut *pitch* terdapat pada Gambar 2.19.



Gambar 2.19 Rotasi *Quadcopter* Sumbu *y*

$$\begin{cases} x_e = x_b \cos \theta + z_b \sin \theta \\ y_e = y_b \\ z_e = -x_b \sin \theta + z_b \cos \theta \end{cases} \quad (2.6)$$

Persamaan 2.6 dapat dibuat dalam bentuk matriks menjadi Persamaan 2.7.

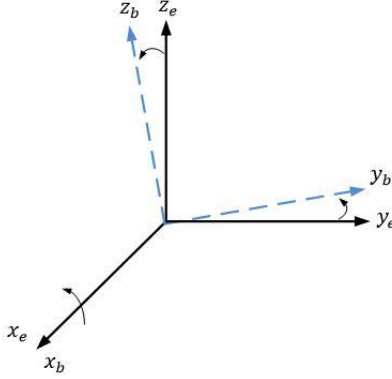
$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \quad (2.7)$$

Adapun matriks rotasi pada sumbu *y* ditunjukkan pada Persamaan 2.8.

$$R(\theta, y) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.8)$$

c. Rotasi Sumbu *x*

Quadcopter berotasi pada sumbu *x* dan menghasilkan sudut *roll* yang dilambangkan dengan $R(\phi, x)$. Rotasi sudut *roll* terdapat pada Gambar 2.20.



Gambar 2.20 Rotasi *Quadcopter* Sumbu x

$$\begin{cases} x_e = x_b \\ y_e = y_b \cos \phi - y_b \sin \phi \\ z_e = y_b \sin \phi + z_b \cos \phi \end{cases} \quad (2.9)$$

Persamaan 2.9 dapat dibuat dalam bentuk matriks menjadi Persamaan 2.10.

$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \quad (2.10)$$

Adapun matriks rotasi pada sumbu z ditunjukkan pada Persamaan 2.11.

$$R(\phi, x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.11)$$

Sehingga persamaan keseluruhan rotasi *Quadcopter* terdapat pada Persamaan 2.12-2.13.

$$R_{\Theta} = R(\phi) R(\theta) R(\psi) \quad (2.12)$$

$$\mathbf{R}_\theta = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta c_\phi & -c_\psi s_\phi + s_\psi s_\theta s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \quad (2.13)$$

Sedangkan untuk kecepatan dalam *Quadcopter* diekspresikan terhadap *body frame* (*b-frame*). Kecepatan *Quadcopter* terdiri dari kecepatan linear V^b dan kecepatan angular ω^b . Komposisi vektornya disajikan dalam Persamaan 2.14-2.15.

$$\omega^b = [p \quad q \quad r]^T \quad (2.14)$$

$$V^b = [u \quad v \quad w]^T \quad (2.15)$$

Diperlukan kombinasi nilai linear dan angular untuk memberikan representasi yang lengkap dalam *space*. ξ merupakan komposisi dari vektor posisi linear Γ^e (m) dan vektor posisi sudut Θ^e (rad) terhadap *earth frame* (*e-frame*) seperti terlihat pada Persamaan 2.16.

$$\xi = [\Gamma^e \quad \Theta^e]^T = [x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T \quad (2.16)$$

Dimana V merupakan generalisasi dari vektor kecepatan linear *Quadcopter* V^b (m s^{-1}) dan kecepatan angular *Quadcopter* ω^b (rad s^{-1}) pada *body frame* (*b-frame*) yang terdapat pada Persamaan 2.17.

$$V = [V^b \quad \omega^b]^T = [u \quad v \quad w \quad p \quad q \quad r]^T \quad (2.17)$$

Hubungan antara kecepatan linear pada *body frame* (*b-frame*) dan salah satu factor pada *earth frame* (*e-frame*) V^e (atau Γ^e) [m/s] dapat dilihat pada Persamaan 2.18.

$$V^e = \Gamma^e = \mathbf{R}_\theta V^b \quad (2.18)$$

Dimana \mathbf{R}_θ adalah matrik rotasi dari *body frame* (*b-frame*) ke *earth frame* (*e-frame*). Seperti pada kecepatan linear, hal tersebut juga berlaku untuk menghubungkan kecepatan angular pada *e-frame* (atau kecepatan Euler) $\dot{\Theta}^e$ [rad/s] ke *b-frame* ω^b atau sebaliknya. Hubungan tersebut terdapat pada Persamaan 2.19 – 2.20.

$$\omega^b = \mathbf{T}_\theta^{-1} \dot{\Theta}^e \quad (2.19)$$

$$\Theta^e = T_\Theta \omega^b \quad (2.20)$$

Dimana T_Θ adalah matriks transformasi. Matriks transformasi T_Θ dapat ditetapkan dengan menggunakan kecepatan Euler dalam *b-frame*, dengan membalik pola perputaran sudut dari *roll*, *pitch* dan *yaw* seperti pada Persamaan 2.21.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R(\phi)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R(\phi)^{-1} R(\theta)^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (2.21)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = T_\Theta^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.22)$$

Persamaan 2.12-2.20 maka diperoleh matriks transformasi dari *body frame* (*b-frame*) menuju *earth frame* (*e-frame*).

$$T_\Theta^{-1} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \quad (2.23)$$

$$T_\Theta = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \quad (2.24)$$

Persamaan yang sudah didapat, maka dibentuk suatu hubungan antara kecepatan terhadap *earth frame* (*e-frame*) dan *body frame* (*b-frame*).

$$\dot{\xi} = J_\Theta v \quad (2.25)$$

Dimana $\dot{\xi}$ adalah vektor kecepatan yang mengacu pada *earth frame* (*e-frame*), v adalah vektor kecepatan mengacu *body frame* (*b-frame*) dan J_Θ adalah matrik *jacobian*. Matrik *jacobian* terdiri dari 4 *sub*-matrik sebagaimana Persamaan 2.26.

$$J_\Theta = \begin{bmatrix} R_\Theta & 0_{3 \times 3} \\ 0_{3 \times 3} & T_\Theta \end{bmatrix} \quad (2.26)$$

2.3.2. Dinamika *Quadcopter* [1]

Dalam pemodelan dinamika *Quadcopter*, maka model dinamis direpresentasikan dengan asumsi penyederhanaan. Ketika *Quadcopter* bergerak secara perlahan efek momentum *Quadcopter* dari pergerakan dapat diabaikan. *Frame* yang digunakan pada *Quadcopter* dikatakan *rigid* dan simetris. Dinamika yang terjadi dipresentasikan dengan persamaan *Euler-Newton* pada Persamaan 2.27.

$$\begin{bmatrix} m I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} \dot{V}^b \\ \dot{\omega}^b \end{bmatrix} + \begin{bmatrix} \omega^b \times (m V^b) \\ \omega^b \times (I \omega^b) \end{bmatrix} = \begin{bmatrix} F^b \\ \tau^b \end{bmatrix} \quad (2.27)$$

Notasi $I_{3 \times 3}$ berarti matrik identitas 3 kali 3. \dot{V}^b [ms⁻²] adalah *vector linier* percepatan *Quadcopter* mengacu *b-frame* sementara $\dot{\omega}^b$ [rads⁻²] adalah vektor percepatan sudut *Quadcopter* mengacu *B-frame*. Selain itu, F^b [N] adalah vektor gaya *Quadcopter* mengacu *b-frame* dan τ^b [N m] adalah vektor torsi *Quadcopter* pada *b-frame*. Hukum Newton kedua tentang gerakan berlaku rumus :

$$F = m \frac{dv}{dt_e} \quad (2.28)$$

Dimana m adalah massa dari sistem yang digunakan, a adalah percepatan gerak, dan F adalah gaya yang terjadi pada sistem. Dari Persamaan *Coriolis* didapatkan Persamaan 2.29 dan Persamaan 2.30.

$$F = m \frac{dv}{dt_e} = m \left(\frac{dv}{dt_b} + \omega \times v \right) \quad (2.29)$$

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \frac{dv}{dt_e} = m \left(\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \right) \quad (2.30)$$

Sehingga percepatan linear dari *Quadcopter* dapat dihitung :

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} - \begin{bmatrix} \dot{\theta} \dot{z} - \dot{\psi} \dot{y} \\ \dot{\psi} \dot{x} - \dot{y} \dot{z} \\ \dot{\phi} \dot{y} - \dot{\theta} \dot{x} \end{bmatrix} \quad (2.31)$$

Untuk menghitung gaya putar yang terjadi pada *Quadcopter* menggunakan hukum Newton kedua seperti pada Persamaan 2.32.

$$M = \frac{dh}{dt_e} \quad (2.32)$$

Dimana M adalah *moment* yang terjadi dan h merupakan momentum putar. Dengan menggunakan persamaan *Coriolis* didapatkan :

$$M = \frac{dv}{dt_e} = \frac{dh}{dt_b} + \omega \times h \quad (2.33)$$

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = I \begin{bmatrix} \ddot{\theta} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\theta} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times I \begin{bmatrix} \dot{\theta} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.34)$$

$$\begin{bmatrix} \ddot{\theta} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = I^{-1} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} - I^{-1} \begin{bmatrix} \dot{\theta} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times I \begin{bmatrix} \dot{\theta} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.35)$$

I merupakan matrik dari momen inersia yang terjadi pada sumbu x_b , y_b , dan z_b yang direpresentasikan kedalam variabel I_{xx} , I_{yy} , dan I_{zz} . Momen inersia adalah ukuran kelembaman suatu benda untuk berotasi terhadap porosnya. Momen inersia berperan dalam dinamika rotasi seperti massa dalam dinamika dasar, dan menentukan hubungan antara momentum sudut dan kecepatan sudut, momen gaya dan percepatan sudut, dan beberapa besaran lain. Maka didapatkan persamaan untuk menghitung moment inersia x , y dan z . I_{xx} , I_{yy} , dan I_{zz} ditunjukkan oleh Persamaan 2.36 sampai 2.38.

$$I_{xx} = \frac{ml^2}{4} + \frac{mh^2}{6} + 2ml^2 + \frac{MR^2}{4} + \frac{MH^2}{12} \quad (2.36)$$

$$I_{yy} = \frac{ml^2}{4} + \frac{mh^2}{6} + 2ml^2 + \frac{MR^2}{4} + \frac{MH^2}{12} \quad (2.37)$$

$$I_{zz} = \frac{MR^2}{2} + 4ml^2 \quad (2.38)$$

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \quad I^{-1} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \quad (2.39)$$

Dengan memasukkan matrik I dan I^{-1} ke dalam Persamaan 2.35 maka akan didapatkan Persamaan 2.40 dan Persamaan 2.41.

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \left(\begin{bmatrix} 0 & \dot{\psi} & -\dot{\theta} \\ -\dot{\psi} & 0 & \dot{\phi} \\ \dot{\theta} & -\dot{\phi} & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \tau_{\phi} \\ \tau_{\theta} \\ \tau_{\psi} \end{bmatrix} \right) \quad (2.40)$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{I_{yy}-I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi} \\ \frac{I_{zz}-I_{yy}}{I_{yy}} \dot{\phi} \dot{\psi} \\ \frac{I_{xx}-I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} \end{bmatrix} + \begin{bmatrix} \frac{1}{I_{xx}} \tau_{\phi} \\ \frac{1}{I_{yy}} \tau_{\theta} \\ \frac{1}{I_{zz}} \tau_{\psi} \end{bmatrix} \quad (2.41)$$

Dan persamaan dinamika dari *Quadcopter* dituliskan pada Persamaan 2.42 dan 2.43.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} - \begin{bmatrix} \dot{\theta} \dot{z} - \dot{\psi} \dot{y} \\ \dot{\psi} \dot{x} - \dot{y} \dot{z} \\ \dot{\phi} \dot{y} - \dot{\theta} \dot{x} \end{bmatrix} \quad (2.42)$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{I_{yy}-I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi} \\ \frac{I_{zz}-I_{yy}}{I_{yy}} \dot{\phi} \dot{\psi} \\ \frac{I_{xx}-I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} \end{bmatrix} + \begin{bmatrix} \frac{1}{I_{xx}} \tau_{\phi} \\ \frac{1}{I_{yy}} \tau_{\theta} \\ \frac{1}{I_{zz}} \tau_{\psi} \end{bmatrix} \quad (2.43)$$

2.4. Logika Fuzzy

Fuzzy berarti kabur atau samar atau tidak jelas. Manusia terbiasa mengolah suatu hal, baik data maupun fakta secara *fuzzy*, bahkan saat pengambilan keputusan didasarkan pada hal-hal yang bersifat *fuzzy*. Konsep logika *fuzzy* berawal dari teori *fuzzy set* sebagai salah satu pendekatan untuk menyelesaikan permasalahan yang tidak memiliki ketentuan yang pasti. Teori tersebut dikembangkan oleh Lotfi Zadeh pada tahun 1965 di University of California – Barkeley. Dalam jurnalnya, Zadeh memperkenalkan konsep teori himpunan baru yang dinamakan *fuzzy set*. Konsep logika *fuzzy* menggantikan konsep “benar-salah” dari logika *boolean* menjadi derajat tingkat kebenaran. Teori *fuzzy* menyatakan keanggotaan suatu objek ke dalam fungsi derajat keanggotaan (*membership function*). Hal tersebut memungkinkan keanggotaan suatu objek dapat dinyatakan pada semua bilangan riil antara

0 sampai 1. Oleh karena itu, konsep *fuzzy* tersebut sesuai dengan pola pikir manusia yang cenderung menilai suatu objek secara samar.

2.4.1. Himpunan *Fuzzy* (*Fuzzy Set*)

Himpunan *Fuzzy* merupakan suatu himpunan yang beranggotakan sejumlah istilah dalam pengertian bahasa yang menyatakan level kualitatif dari semesta pembicaraan. Misalnya pengukuran tekanan gas dapat diterjemahkan ke dalam beberapa istilah bahasa yang menyatakan level kualitatif dari besar tekanan gas tersebut, sehingga apabila semesta pembicaraan berupa besar tekanan gas, maka dapat dibuat suatu himpunan *fuzzy* yaitu "Sangat Kecil", "Kecil", "Sedang", "Besar", "Sangat Besar".

2.4.2. Fungsi Keanggotaan (*Membership Function*)

Fungsi keanggotaan (*membership function*) adalah kurva yang menunjukkan pemetaan input atau output data ke dalam nilai keanggotaannya dengan rentang nol sampai satu. Berikut merupakan penjelasan beberapa bentuk fungsi keanggotaan yang umum digunakan, yaitu :

a. Fungsi Segitiga

Fungsi segitiga merupakan gabungan dua fungsi linear. Fungsi ini ditentukan oleh tiga parameter $\{a, b, c\}$ dengan ketentuan $a < c < b$. Fungsi keanggotaan fungsi segitiga ini yaitu :

$$\mu(x) = \begin{cases} 0 & ; x \leq a \\ \frac{x-a}{c-a} & ; a \leq x \leq c \\ \frac{b-x}{b-c} & ; c \leq x \leq b \\ 0 & ; x \geq b \end{cases} \quad (2.44)$$

b. Fungsi Trapesium

Fungsi trapesium pada dasarnya menyerupai fungsi segitiga dengan nilai keanggotaan satu. Fungsi ini ditentukan oleh empat parameter $\{a, b, c, d\}$ dengan ketentuan $a < c \leq d < b$. Fungsi keanggotaan trapesium ini yaitu :

$$\mu(x) = \begin{cases} 0; x \leq a \\ \frac{x-a}{c-a}; a \leq x \leq c \\ 1; c \leq x \leq d \\ \frac{b-x}{b-d}; d \leq x \leq b \\ 0; x \geq b \end{cases} \quad (2.45)$$

c. Fungsi Gaussian

Fungsi gaussian menggunakan dua parameter, yaitu (c) untuk menunjukkan nilai domain pada pusat kurva, dan (rho) menunjukkan lebar fungsi. Fungsi keanggotaan *gaussian* ini yaitu :

$$\mu(x) = \left\{ e^{-\left(\frac{x-c}{a}\right)^2} \right. \quad (2.46)$$

d. Fungsi S

Fungsi S didefinisikan dengan menggunakan tiga parameter, yaitu nilai keanggotaan nol, nilai keanggotaan penuh, dan titik infleksi. Titik infleksi yaitu titik yang memiliki domain 50% benar. Fungsi keanggotaan fungsi S, yaitu :

$$\mu(x) = \begin{cases} 0; x \leq a \\ 2 \left(\frac{x-a}{y-a} \right)^2; a \leq x \leq \beta \\ 1 - 2 \left(\frac{x-a}{y-a} \right)^2; \beta \leq x \leq \gamma \\ 1; x \geq \gamma \end{cases} \quad (2.47)$$

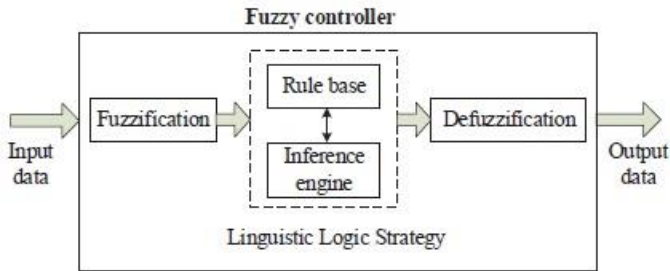
e. Fungsi Bell

Fungsi bell berbentuk lonceng, menggunakan dua parameter yaitu (c) untuk menunjukkan derajat keanggotaan bernilai satu yang terletak di pusat himpunan, dan lebar fungsi (b). Fungsi keanggotaan fungsi bell yaitu:

$$\mu(x) = \begin{cases} S \left(x; y - \beta, \gamma - \frac{\beta}{2}, \gamma \right); x \leq \gamma \\ 1 - S \left(x; y + \beta, \gamma + \frac{\beta}{2}, \gamma \right); x > \gamma \end{cases} \quad (2.48)$$

2.4.3. Kontroler Logika Fuzzy [11]

Kontroler logika *fuzzy* merupakan suatu kontroler dimana proses perhitungan sinyal kontrol melalui operasi himpunan *fuzzy* meliputi proses *fuzzifikasi*, relasi *fuzzy*, inferensi *fuzzy* serta *defuzzifikasi* seperti terlihat pada Gambar 2.21.



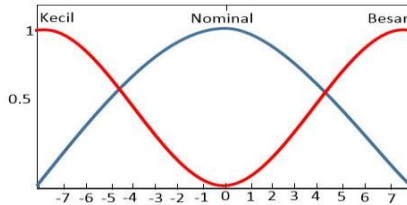
Gambar 2.21 Kontrol Logika Fuzzy

Proses di dalam himpunan *fuzzy* pada kontroler *fuzzy* antara lain sebagai berikut :

a. *Fuzzifikasi*

Fuzzifikasi adalah proses transformasi nilai real (nilai bukan *fuzzy*) menjadi nilai suatu himpunan *fuzzy* yang dinyatakan dalam derajat keanggotaan yang didefinisikan dengan salah satu fungsi keanggotaan. Struktur fungsi keanggotaan *fuzzy* dapat dilihat pada Gambar 2.22. Derajat keanggotaan adalah derajat dari masukkan tegas pada sebuah fungsi keanggotaan, memiliki nilai 0 s/d 1. Fungsi keanggotaan mendefinisikan nilai *fuzzy* dengan melakukan pemetaan nilai tegas berdasarkan daerahnya untuk diasosiasikan dengan derajat keanggotaan.

Masukkan tegas pada umumnya merupakan hasil pengukuran parameter eksternal dari sistem kontrol. Label merupakan deskripsi nama untuk menunjukkan suatu fungsi keanggotaan *fuzzy* yang memiliki domain (lebar fungsi keanggotaan *fuzzy*) tertentu. Semesta pembicaraan memiliki jarak yang mencakup seluruh masukkan tegas yang mungkin ada. Bentuk fungsi keanggotaan harus mewakili variabel masukkan tegas, namun bentuk yang digunakan dibatasi oleh kemampuan *tool* dalam melakukan perhitungan. Bentuk fungsi yang rumit membutuhkan persamaan fungsi yang lebih kompleks.



Gambar 2.22 Contoh fuzzifikasi dengan 3 anggota himpunan

b. Basis Aturan

Basis aturan merupakan deskripsi linguistik terhadap variabel input dan output. Penentuan basis aturan yang dipakai dalam mengontrol suatu *plant* dapat melalui metode heuristik maupun deterministik. Metode heuristik didasarkan pada pengetahuan terhadap *plant* dan perilaku dari *plant* yang akan dikontrol. Sedangkan metode deterministik diperoleh melalui identifikasi struktur dan parameter dari aturan kontrol. Pemetaan input dan output pada sistem *fuzzy* direpresentasikan dalam pernyataan berikut : **IF** premis **THEN** konsekuen.

Pada kontroler *fuzzy*, input kontroler direpresentasikan dalam premis dan output kontroler direpresentasikan dalam konsekuen. Jumlah basis aturan dari suatu sistem *fuzzy* ditentukan dari jumlah variabel pada input dan jumlah *membership function* pada variabel input, dirumuskan dalam Persamaan 2.49.

$$\prod_{i=1}^n N_i = N_1 \times N_2 \times N_3 \times \dots \times N_n \quad (2.49)$$

Dimana N_i merupakan jumlah *membership function* pada variabel input i . Sebagai contoh apabila variabel input pertama memiliki tujuh *membership function* dan variabel input kedua memiliki tujuh *membership function*, maka jumlah basis aturan adalah $7 \times 7 = 49$ aturan.

c. Mekanisme Inferensi

Mekanisme inferensi pada kontroler *fuzzy* merupakan mekanisme operasi matematika yang dilakukan sesuai dengan premis yang diberikan. Pada basis aturan terdapat dua operasi matematika yang dapat dilakukan yaitu *AND* dan *OR*. Terdapat beberapa tipe mekanisme inferensi *fuzzy* yaitu metode Mamdani, Larsent, dan Takagi sugeno. Perbedaan dari metode ini terletak

pada pengambilan kesimpulan logika *fuzzy*. Pada metode Mamdani maupun Larsent, kesimpulan logika *fuzzy* berupa derajat keanggotaan sehingga dalam menyimpulkan suatu logika *fuzzy* dibutuhkan proses defuzzifikasi. Sedangkan pada tipe Takagi Sugeno, kesimpulan logika *fuzzy* berupa suatu persamaan sehingga tidak diperlukan proses defuzzifikasi. Kelebihan pada logika *fuzzy* tipe Mamdani dan Larsent lebih sederhana, akan tetapi diperlukan kemampuan untuk mengetahui karakteristik plant untuk menentukan batasan keluaran kontroler. Pada tipe Takagi-Sugeno tidak diperlukan pengetahuan mengenai karakteristik dari plant akan tetapi diperlukan perhitungan yang lebih rumit untuk persamaan pada bagian konsekuen.

d. *Defuzzifikasi*

Defuzzifikasi adalah suatu proses mentransformasikan harga fuzzifikasi yang telah di inferensi kedalam harga bukan *fuzzy* atau harga aktual. Beberapa metode dalam proses defuzzifikasi yaitu *Center of Area*, *Mean of Maxima*, *weighted sum*, dan lainnya.

1. *Center of Area*

Metode *center of area* digunakan untuk menentukan nilai titik tengah area yang merupakan titik pusat massa dari kombinasi fungsi-fungsi keanggotaan. Secara umum, persamaan untuk metode *Center of Area* ditunjukkan dengan persamaan 2.50

$$U_0 = \frac{\sum_{k=1}^m U_k(T) \cdot \mu_k(U_k(T))}{\sum_{k=1}^m \mu_k(U_k(T))}, \forall u \in U(T) \quad (2.50)$$

2. *Mean of Maxima*

Metode *mean of maxima* mengambil semua nilai tiap fungsi keanggotaan dengan derajat keanggotaan maksimum dan menghitung rata-rata dari nilai-nilai tersebut sebagai keluaran tegas. Persamaan 2.51 menunjukkan persamaan umum metode tersebut.

$$U_0 = \frac{\sum_n \max(\mu_A^n) \cdot y_n}{\sum_n \max(\mu_A^n)} \quad (2.51)$$

3. *Weighted Sum*

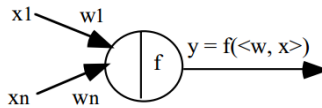
Metode *weighted sum* merupakan metode *defuzzifikasi* dengan fungsi keanggotaan berupa fungsi *singleton*.

Persamaan 2.52 menunjukkan persamaan *defuzzifikasi* fungsi *singleton*.

$$U_o = \frac{\sum_i \mu_A^i s_i}{\sum_i \mu_A^i} \quad (2.52)$$

2.5. Jaringan Saraf Tiruan

Jaringan saraf tiruan merupakan suatu metode komputasi yang merupakan representasi bentuk penyederhanaan dari jaringan sistem saraf biologis yang dinyatakan ke dalam formulasi matematik jaringan simbolik.[10] Gambar 2.23 menunjukkan struktur model jaringan saraf tiruan. Jaringan saraf tiruan dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh atau masukkan dan membuat prediksi tentang kemungkinan keluaran yang akan muncul atau menyimpan karakteristik masukkan yang diberikan kepada jaringan saraf tiruan.

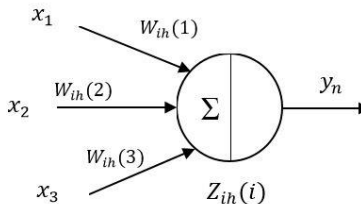


Gambar 2.23 Struktur model jaringan saraf tiruan

Pada jaringan saraf tiruan terdapat dua formulasi, yaitu formulasi *forward* dan formulasi *backward*. Berikut merupakan penjelasan dari kedua formulasi tersebut :

a. formulasi *forward*

formulasi *forward* digunakan untuk menghitung keluaran dari setiap neuron pada jaringan saraf tiruan. Gambar 2.24 menunjukkan struktur formulasi *forward* pada jaringan saraf tiruan yang terdiri dari input, bobot, fungsi aktivasi dan output.



Gambar 2.24 Struktur formulasi *forward*

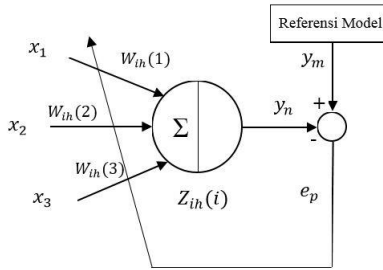
Untuk formulasi *forward* ini menggunakan persamaan 2.53 dan 2.54.

$$Z_{ih}(i) = \sum_{j=1}^n w_{ih}(i, j) x(i) \quad (2.53)$$

$$y_{ih}(i) = Z_{ih} x \lambda \quad (2.54)$$

b. formulasi *backward*

Formulasi *backward* digunakan untuk merevisi bobot dari nilai error yang diperoleh dari proses adaptasi jaringan terhadap keluaran model yang diinginkan. Setiap ada error baru, jaringan dapat belajar dari error tersebut dengan merevisi nilai bobot untuk menyesuaikan karakter nilai. Gambar 2.25 menunjukkan struktur formulasi *backward* pada jaringan saraf tiruan.



Gambar 2.25 Struktur formulasi *Backward*

Persamaan 2.55 dan 2.56 menunjukkan formulasi revisi bobot pada proses *backward*.

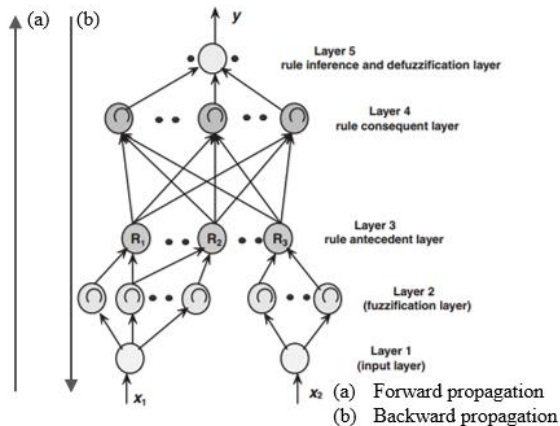
$$e = y_m - y_n \quad (2.55)$$

$$w^b = w^l + \alpha x e_p x(i) \quad (2.56)$$

2.6. Kontroler *Neurofuzzy* [9]

Suatu sistem *hybrid* yang menggabungkan logika *fuzzy*, jaringan saraf tiruan, algoritma genetika, dan kecerdasan buatan lainnya sudah banyak dikembangkan untuk menyelesaikan berbagai masalah yang berkaitan dengan optimisasi. Konsep *hybrid* dikembangkan karena pada metode kecerdasan buatan ini memiliki kelebihan dan kekurangan masing-masing, sehingga dengan konsep *hybrid* akan diperoleh suatu metode yang lebih baik dalam menyelesaikan permasalahan optimasi.

Gambar 2.26 menunjukkan skema atau model dari kontroler *Neurofuzzy*. Terdapat beberapa skema atau struktur *Neurofuzzy*, seperti *Neurofuzzy Mamdani* atau Larsent, *Neurofuzzy TakagiSugeno* (ANFIS / *Artificial Neuro fuzzy Inference System*), *Neurofuzzy Biner* (Multi input) atau *Neurofuzzy* formulasi Wang. Dari Gambar 2.24 dan 2.25 dapat diketahui bahwa dalam struktur *Neurofuzzy* terdapat 2 tahapan yaitu *forward propagation* dan *backward propagation*.



Gambar 2.26 Struktur kontroler *Neurofuzzy* metode mamdani

Kontroler *Neurofuzzy* merupakan pengembangan sistem *hybrid* yang menggabungkan konsep logika *fuzzy* dengan jaringan saraf tiruan (JST). Logika *fuzzy* memiliki kelebihan dalam pengambilan keputusan, sedangkan JST memiliki kelebihan dalam adaptasi melalui kemampuan pembelajaran yang dimilikinya. Kekurangan logika *fuzzy* dalam hal penentuan parameter yang sifatnya intuitif dapat diatasi dengan JST. Pada kontroler *Neurofuzzy*, logika *fuzzy* akan direpresentasikan dalam JST yang memungkinkan terjadinya pembelajaran di dalamnya dan akan terjadi perubahan bobot untuk mengubah parameter pada logika *fuzzy*.

2.6.1 Tahap *Training*

Pendekatan pemodelan dalam fungsi *Neurofuzzy* serupa dengan teknik-teknik identifikasi sistem pada umumnya. Pertama *Neurofuzzy* mengasumsikan adanya sebuah struktur tertentu yang menghubungkan input dengan output. Kemudian *Neurofuzzy* harus diberikan pasangan

data input dan output dalam format yang kompatibel untuk *training*. Jika kedua tahap terpenuhi maka *Neurofuzzy* siap digunakan untuk melatih FIS sehingga mampu menirukan kelakuan sistem yang sedang dimodelkan. *Neurofuzzy* melatih FIS dengan memodifikasi parameter-parameter fungsi keanggotaan sampai diperoleh selisih (*error*) minimal antara keluaran FIS dengan data pelatihan output.

Pola *training* yang dilakukan untuk *Quadcopter* ada berbagai macam cara seperti menjalankan dengan menggunakan *remote control* kemudian data kecepatan dan pembacaan sensor disimpan untuk dijadikan data *training* atau dengan menggerakkan *Quadcopter* pada tiap-tiap sumbu lintasannya untuk mendapatkan data pembacaan sensor.

2.6.2 Tahap *Forward Propagation*

Tahapan *forward propagation* merupakan tahapan perhitungan sistem *fuzzy* yang direpresentasikan ke dalam beberapa layer seperti dalam *neural network*. Tahapan ini bertujuan untuk menentukan *output layer* dari suatu *node*. Dalam sistem ini terdapat 3 lapis data yang dalam beberapa lapisan diantaranya *input layer*, *hidden layer*, dan *output layer*.

a. *Input layer*

Input layer merupakan *node* masukkan yang mengirim sinyal masukkan ke *hidden layer*. Pada lapisan ini nilai keluaran *node* merupakan hasil dari fuzzifikasi *input variable* sistem. Lapisan ini berfungsi sebagai *membership function* untuk mengekspresikan nilai linguistik dari variabel linguistik *input*.

b. *Hidden layer*

Hidden layer digunakan untuk basis aturan (*rule-base*) *fuzzy*, setiap *node* pada lapisan mempunyai masukkan dari dua nilai pada lapisan kedua. Operasi pada lapisan ini berbeda-beda tergantung dari struktur *Neurofuzzy* yang digunakan. Misalnya jika digunakan struktur *Neurofuzzy* Mamdani maka digunakan operasi MIN sedangkan jika menggunakan *Neurofuzzy* Larsent digunakan operasi *product*. Persamaan 2.57 menunjukkan persamaan basis aturan (*rule-base*) pada struktur *Neurofuzzy* mamdani.

$$R_{i,4} = \text{Min} (O_{n,1}, O_{m,1}) \text{ untuk } i = 1, 2, \dots, n \times m \quad (2.57)$$

Pada lapisan ini juga menjalankan proses inferensi sehingga keluaran pada *node* ini diperoleh dari operasi inferensi. Sama

halnya pada proses sebelumnya, pada lapisan ini dilakukan operasi inferensi yang bergantung pada struktur *Neurofuzzy* yang digunakan. Misalnya jika digunakan struktur *Neurofuzzy* Mamdani maka digunakan operasi MAX sedangkan jika menggunakan *Neurofuzzy* Larsent digunakan operasi jumlah. Persamaan 2.15 menunjukkan operasi MAX pada struktur *Neurofuzzy* mamdani.

$$r_{i,4} = \text{Max} (w_{i,j}, R_{i,4}) \text{ untuk } i = 1,2 \dots, n \times m \quad (2.58)$$

Pada struktur *Neurofuzzy* mamdani/larsent, bobot Wij bernilai satu karena pada struktur ini tidak terdapat revisi nilai bobot pada proses inferensi. Revisi bobot Wij pada lapisan ini dilakukan apabila menggunakan struktur *Neurofuzzy* takagi-sugeno, dimana revisi bobot digunakan untuk merevisi rule base *fuzzy*.

c. *Output layer*

Output layer merupakan lapisan yang menjalankan proses *defuzzikasi* untuk menghitung sinyal keluaran dari kontroler *Neurofuzzy*. Apabila digunakan fungsi *fuzzy singleton* maka persamaan 2.59 menunjukkan persamaan dari fungsi *center of gravity*. Dengan bobot adalah w_i yang merupakan nilai tengah dari fungsi keanggotaan sinyal keluaran.

$$y = \frac{\sum_{i=1}^n w_i x U y_i}{\sum_{i=1}^n U y_i} \quad (2.59)$$

2.6.3 Tahap *Backward Propagation*

Pada kontroler *Neurofuzzy*, proses *backward* dilakukan untuk merevisi nilai tengah dari *membership function*. Proses tersebut terdapat pada struktur *Neurofuzzy* Mamdani atau Larsent. [5] Proses *backward* hanya dilakukan satu kali yaitu untuk merevisi nilai tengah dari fungsi keanggotaan sinyal kontrol saja. Berbeda dengan struktur *Neurofuzzy* Takagi-sugeno (ANFIS), proses *backward* dilakukan sampai ke proses inferensi. Jadi proses *backward* tidak hanya merevisi bobot sinyal kontrol tetapi juga merevisi *rule base* pada proses inferensi. Persamaan 2.60 menunjukkan persamaan revisi nilai tengah pada struktur *Neurofuzzy* Mamdani.

$$w^B = w^l(i) + \alpha x e_{num} x \lambda + U y(i) \quad (2.60)$$

--halaman ini sengaja dikosongkan--

BAB III

PERANCANGAN SISTEM

Pada bagian ini akan dibahas mengenai perancangan sistem *Quadcopter* meliputi spesifikasi, identifikasi kebutuhan, pemodelan *Quadcopter*, identifikasi konstanta, perancangan kontroler *Neurofuzzy*, dan simulasi sistem telusur trajektori.

3.1. Desain Acuan *Quadcopter* Pada Simulasi

Spesifikasi desain *Quadcopter* merupakan hal yang penting saat melakukan perancangan sistem. Spesifikasi desain yang harus dipenuhi salah satunya yaitu kerangka (*frame*) *Quadcopter*. Pemilihan konfigurasi *frame Quadcopter* harus ditentukan dahulu. Pada penelitian ini digunakan *frame* dengan konfigurasi “*plus(+)*”. *Frame* tersebut harus dibuat seringan mungkin dan simetris. Dengan desain kerangka yang simetris ini memudahkan dalam pemodelan *Quadcopter* pada tahap simulasi.

Desain yang rigid dan simetris akan mendukung pergerakan *Quadcopter*. *Quadcopter* yang dikatakan simetris dapat dilihat dari nilai momen inersia pada translasi maju dan menyamping yang memiliki nilai sama. Desain mekanik *Quadcopter* pada penelitian ini dapat dilihat pada Gambar 3.1. Desain *Quadcopter* tersebut yang akan dijadikan acuan dalam perancangan simulasi.



Gambar 3.1 Desain acuan *Quadcopter*

Kerangka yang digunakan pada penelitian ini dibuat dari bahan *carbon fiber* yang ringan. Perancangan *Quadcopter* ini berbentuk plus “+” dengan spesifikasi :

Dimensi	: 461x103x328 mm
Motor Bolt Holes	: 22 - 32 mm
Berat <i>frame</i>	: 198 gram
<i>Mounting</i>	: 4 ESC, 4 Motor <i>Brushless</i> , dan 4 <i>Propeller</i>

3.2. Identifikasi Konstanta

Model matematik yang diperoleh dengan pemodelan yang telah dijelaskan pada bab dua, maka terlihat bahwa dibutuhkan konstanta-konstanta yang terdapat pada sistem agar model matematik bisa digunakan pada simulasi. Beberapa konstanta yang harus diperoleh antara lain yaitu, massa (m), momen inersia (I_{xx} , I_{yy} , I_{zz} , dan I_r), pemetaan gaya angkat terhadap pulsa, konstanta *thrust* (b), dan konstanta *drag* (d).

Massa dari sistem keseluruhan diukur menggunakan alat pengukur massa. Sedangkan untuk momen inersia dari sistem terhadap sumbu x , sumbu y , dan sumbu z diperoleh dengan perhitungan fisis. Dengan spesifikasi motor dan *propeller* seperti yang tertera pada *datasheet* motor, diperoleh konstanta *drag* dan konstanta *thrust*.

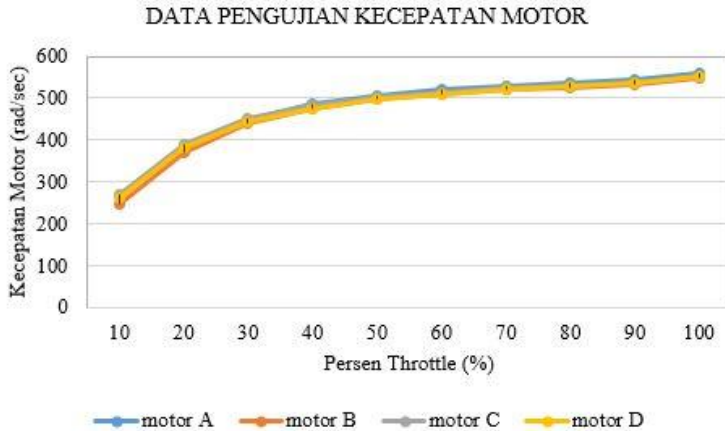
3.2.1. Pengukuran Kecepatan Motor dan *Propeller*

Untuk mengetahui model linier dari kecepatan motor maka dilakukan pengukuran menggunakan *tachometer*. Motor diputar secara bertahap dari kecepatan rendah sampai kecepatan tertinggi. Input pulsa di berikan melalui program *mission planner*. Tabel 3.1 menunjukkan hasil pembacaan kecepatan motor.

Tabel 3.1 Hasil pengukuran kecepatan motor *propeller*.

%Throtlle	Throttle	Motor 1 (rad/s)	Motor 2 (rad/s)	Motor 3 (rad/s)	Motor 4 (rad/s)
10	1092	265	246	271	262
20	1195	386	371	390	380
30	1297	449	440	450	442
40	1400	485	475	483	476
50	1502	507	496	501	497
60	1604	520	510	514	511
70	1707	530	519	523	520
80	1809	537	526	530	527
90	1912	544	533	536	535
100	2014	561	549	551	550

Berdasarkan hasil pengukuran pada Tabel 3.1 dijadikan grafik untuk memperjelas hubungan antara pulsa *input* yang diberikan pada motor dan kecepatan motor yang telah dikonversi dalam satuan *rad/sec*. Grafik tersebut ditunjukkan pada Gambar 3.2.



Gambar 3.2 Hubungan kecepatan motor dan PWM

Dari hasil pengukuran diperoleh persamaan linier hubungan pulsa motor terhadap kecepatan (rad/s) pada masing-masing motor adalah :

$$y_1 = 2.6446x_1 + 332.88 \quad (3.1)$$

$$y_2 = 2.6911x_2 + 318.44 \quad (3.2)$$

$$y_3 = 2.468x_3 + 339.14 \quad (3.3)$$

$$y_4 = 2.5757x_4 + 328.38 \quad (3.4)$$

3.2.2. Pengukuran Gaya Angkat Motor dan *propeller*

Langkah selanjutnya adalah pengukuran gaya angkat *propeller*. Pada pengukuran ini dilakukan dengan meletakkan *propeller* diatas beban yang sudah diukur dengan timbangan. Kemudian motor diputar secara kontinyu dari kecepatan rendah sampai tertinggi (20 – 100 %) sesuai presentasi dari nilai PWM pada *ground station*. Nilai yang muncul pada tampilan timbangan kemudian dicatat sebagai hasil dari tiap-tiap persentase nilai PWM yang diberikan. Hasil pengukuran gaya angkat motor terhadap pulsa ditunjukkan pada Tabel 3.2.

Tabel 3.2 Hasil pengukuran gaya angkat motor *propeller*.

Pulse %	Motor 1 (gram)	Motor 2 (gram)	Motor 3 (gram)	Motor 4 (gram)
20	58	60	60	60
30	91	94	92	94
50	183	183	185	181
60	250	236	239	240
70	303	298	300	301
100	446	448	447	447

Dari pengukuran diperoleh persamaan linier hubungan pulsa motor terhadap gaya angkat (gram) pada masing-masing motor adalah :

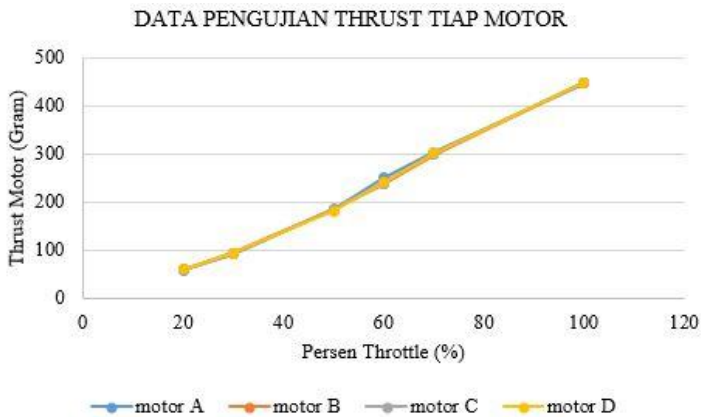
$$y_1 = 4.9747x_1 - 51.775 \quad (3.5)$$

$$y_2 = 4.9265x_2 - 51.124 \quad (3.6)$$

$$y_3 = 4.9361x_3 - 50.988 \quad (3.7)$$

$$y_4 = 4.9337x_4 - 50.855 \quad (3.8)$$

Berdasarkan hasil pengukuran pada Tabel 3.2 dijadikan grafik untuk memperjelas hubungan antara pulsa *input* dan gaya angkat yang diberikan pada motor dalam satuan gram. Grafik tersebut ditunjukkan pada Gambar 3.3 berikut.



Gambar 3.3 Hubungan kecepatan motor dan gaya angkat

Sehingga, besar gaya angkat (Kg) pada tiap-tiap motor pada penelitian ini adalah :

$$\tau_1 = 9.81 * (4.9747x - 51.775)/1000 \quad (3.9)$$

$$\tau_2 = 9.81 * (4.9265x - 51.124)/1000 \quad (3.10)$$

$$\tau_3 = 9.81 * (4.9361x - 50.988)/1000 \quad (3.11)$$

$$\tau_4 = 9.81 * (4.9337x - 50.855)/1000 \quad (3.12)$$

3.2.3. Pengukuran Gaya Thrust

Untuk mendapatkan konstanta *thrust* yaitu ketika menerbangkan *Quadcopter* pada posisi *hover*. Hal ini dikarenakan pada posisi *hover* gaya yang dihasilkan oleh motor untuk mengangkat adalah sama dengan gaya ke bawah. Dengan asumsi gaya gravitasi tetap, maka konstanta *thrust* dapat dihitung menggunakan persamaan 3.13.

$$b = \frac{m \cdot g}{\sum_{i=0}^n \Omega_i^2} \quad (3.13)$$

Berdasarkan data tersebut didapatkan kecepatan tiap-tiap motor ketika dalam keadaan *hover*. *Quadcopter* yang dibuat memiliki masa total 1,2 Kg dan konstanta gravitasi adalah 9,8 ms⁻². Dengan memasukkan data-data tersebut ke persamaan maka didapatkan konstanta *thrust* sebesar 2,2478x10⁻⁶.

3.2.4. Pengukuran Gaya Drag

Konstanta *drag* dihitung dengan persamaan gerak lurus berubah beraturan. Gerak lurus berubah beraturan adalah gerak yang lintasannya berupa garis lurus dengan kecepatannya yang berubah beraturan. Jadi pengukuran konstanta *drag* dilakukan dengan mengambil data terbang *Quadcopter* pada saat *Quadcopter take-off*. Gaya yang terjadi saat *Quadcopter* bergerak ke atas adalah :

$$\sum F = Thrust - gravitasi - Drag \quad (3.14)$$

$$ma = (\tau_1 + \tau_2 + \tau_3 + \tau_4) - mg - D \quad (3.15)$$

Sehingga :

$$D = (\tau_1 + \tau_2 + \tau_3 + \tau_4) - mg - ma \quad (3.16)$$

Dimana :

$$a = \frac{s - v_0 \cdot t}{0,5 \cdot t^2} \quad (3.17)$$

dan

$$CD = \frac{D}{\Omega^2} \quad (3.18)$$

Dengan ketentuan :

- CD = Konstanta *drag*
- v_o = Kecepatan awal (m/detik)
- t = Waktu (detik)
- a = Percepatan (m/detik²)
- s = Jarak yang ditempuh (m)
- m = Massa *Quadcopter* (kg)
- τ_i = Gaya angkat (N/m), $i=1,2,3,4$
- D = Gaya *drag*

3.2.5. Perhitungan Konstanta Inersia

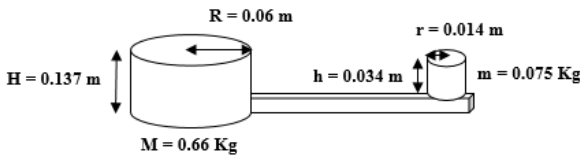
Momen inersia berperan dalam dinamika rotasi seperti massa dalam dinamika dasar, dan menentukan hubungan antara momentum sudut dan kecepatan sudut, momen gaya dan percepatan sudut, dan beberapa besaran lain. Perhitungan nilai parameter I_{xx} , I_{yy} , dan I_{zz} dengan mengasumsi bentuk dari tengah *Quadcopter* adalah silinder dan bentuk dari motor juga silinder. Persamaan yang digunakan untuk menghitung I_{xx} , I_{yy} , dan I_{zz} ditunjukkan oleh Persamaan 3.19 - 3.21.

$$I_{xx} = \frac{mr^2}{4} + \frac{mh^2}{6} + 2mr^2 + \frac{MR^2}{4} + \frac{MH^2}{12} \quad (3.19)$$

$$I_{yy} = \frac{mr^2}{4} + \frac{mh^2}{6} + 2mr^2 + \frac{MR^2}{4} + \frac{MH^2}{12} \quad (3.20)$$

$$I_{zz} = \frac{MR^2}{2} + 4mr^2 \quad (3.21)$$

Untuk mendapatkan nilai I_{xx} , I_{yy} , dan I_{zz} , kita harus mengetahui massa, tinggi, dan jari-jari yang ada dikedua silinder pada *Quadcopter*.



Gambar 3.4 Massa, Tinggi, dan Jari-jari *Quadcopter*

Untuk mengetahui nilai momen inersia pada motor *Quadcopter*, maka digunakan rumus pada Persamaan 3.22.

$$J_r = \frac{4 \times (m \times r^2)}{2} \quad (3.22)$$

Dengan memasukkan nilai yang telah diketahui sesuai Gambar 3.4 ke dalam persamaan I_{xx} , I_{yy} , dan I_{zz} , maka bisa di dapatkan nilai parameter I_{xx} adalah $1,680157 \times 10^{-3} \text{ Kg m}^2$, I_{yy} adalah $1,680157 \times 10^{-3} \text{ Kg m}^2$, I_{zz} adalah $1,20466 \times 10^{-3} \text{ Kg m}^2$ dan momen inersia motor adalah $0,025872 \text{ Kg m}^2$. Pada *Quadcopter* dianggap tidak ada yang berbentuk *aerodinamis* sehingga gaya *aerodinamis* dan momen *aerodinamis* yang terjadi dapat diabaikan.

Dari gaya-gaya yang terjadi pada tiap motor di *Quadcopter*, dapat dihitung persamaan torsi yang terjadi pada *roll*, *pitch*, dan *yaw*. Persamaan tersebut dapat dilihat pada persamaan 3.23 sampai 3.26.

$$U_1 = F_f + F_r + F_b + F_l = b(\Omega_f^2 + \Omega_r^2 + \Omega_b^2 + \Omega_l^2) \quad (3.23)$$

$$U_2 = bl(-\Omega_r^2 + \Omega_l^2) \quad (3.24)$$

$$U_3 = bl(\Omega_f^2 - \Omega_b^2) \quad (3.25)$$

$$U_4 = d(-\Omega_f^2 + \Omega_r^2 - \Omega_b^2 + \Omega_l^2) \quad (3.26)$$

3.3. Model Matematika Hasil Identifikasi

Dari analisis kinematika dan dinamika yang telah dibahas pada bab dua, diperoleh persamaan model matematika dari *Quadcopter* secara keseluruhan di tunjukkan pada persamaan 3.27-3.32.

$$\ddot{x} = (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi) \frac{U_1}{m} \quad (3.27)$$

$$\ddot{y} = (-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi) \frac{U_1}{m} \quad (3.28)$$

$$\ddot{z} = -g + (\cos\theta\cos\phi) \frac{U_1}{m} \quad (3.29)$$

$$\dot{p} = \frac{I_{yy}-I_{zz}}{I_{xx}} q r - \frac{J_r}{I_{xx}} q \Omega + \frac{U_2}{I_{xx}} \quad (3.30)$$

$$\dot{q} = \frac{I_{zz}-I_{xx}}{I_{yy}} p r + \frac{J_r}{I_{yy}} p \Omega + \frac{U_3}{I_{yy}} \quad (3.31)$$

$$\dot{r} = \frac{I_{xx}-I_{yy}}{I_{zz}} p q + \frac{U_4}{I_{zz}} \quad (3.32)$$

Dengan melihat secara sederhana pada Persamaan 3.23, posisi pada sumbu z , dan posisi sudut *roll*, *pitch*, *yaw* dapat dikontrol secara langsung, berturut-turut dengan menggunakan U_1 , U_2 , U_3 , dan U_4 . Pengaturan pada posisi maju (x), dan menyamping (y) dapat dilakukan dengan mengatur

sudut *pitch* dan *roll* dengan syarat gaya angkat (U_i) tidak sama dengan nol.

Nilai *input* dari *Quadcopter* merupakan gaya angkat tiap *propeller* yang dimodelkan secara teoritis adalah sebagai berikut:

$$U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (3.33)$$

$$U_2 = bl(-\Omega_2^2 + \Omega_4^2) \quad (3.34)$$

$$U_3 = bl(\Omega_3^2 - \Omega_1^2) \quad (3.35)$$

$$U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \quad (3.36)$$

$$\Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \quad (3.37)$$

dimana,

Ω_1 = kecepatan putar motor ke-1

Ω_2 = kecepatan putar motor ke-2

Ω_3 = kecepatan putar motor ke-3

Ω_4 = kecepatan putar motor ke-4

Ω = kecepatan putar total

b = konstanta *thrust*

l = lebar *frame*

d = konstanta *drag* dari *quadrotor*

Dari model matematik yang diperoleh dengan pemodelan fisis, maka konstanta yang diperoleh untuk menentukan parameter *plant Quadcopter* akan ditampilkan pada Tabel 3.3.

Tabel 3.3 Identifikasi konstanta

Konstanta	Nilai	Satuan
Massa <i>Quadcopter</i>	1,2	Kg
Momen inersia rotasi sumbu x (I_{xx})	$1,680158 \times 10^{-3}$	Kgm ²
Momen inersia rotasi sumbu y (I_{yy})	$1,680158 \times 10^{-3}$	Kgm ²
Momen inersia rotasi sumbu z (I_{zz})	$1,20466 \times 10^{-3}$	Kgm ²
Momen Inersia motor-propeler (J_{TP})	0,025872	Kgm ²
Konstanta <i>Thrust</i>	$2,2478 \times 10^{-6}$	N.sec ²
Konstanta <i>Drag</i>	$2,5 \times 10^{-7}$	Nm.sec ²

Setelah diperoleh konstanta-konstanta dari sistem, maka dapat dituliskan kembali model matematika dari sistem *Quadcopter* adalah sebagai berikut:

$$\ddot{x} = (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi) \frac{U_1}{1,2} \quad (3.38)$$

$$\ddot{y} = (-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi) \frac{U_1}{1,2} \quad (3.39)$$

$$\ddot{z} = -9,81 + (\cos\theta\cos\phi) \frac{U_1}{1,2} \quad (3.40)$$

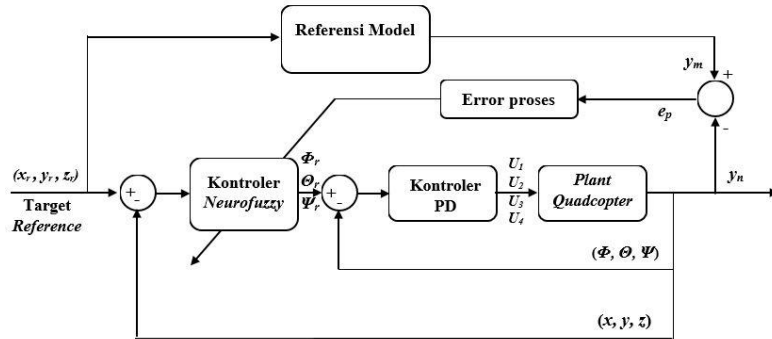
$$\dot{p} = 0,2830qr - 15.3986q\Omega + 595.1821 U_2 \quad (3.41)$$

$$\dot{q} = -0.2830pr + 15.3986p\Omega + 595.1821 U_3 \quad (3.42)$$

$$\dot{r} = 0.00012pq + 830.1097 U_4 \quad (3.43)$$

3.4. Perancangan Kontroler

Setelah diperoleh model matematika dan konstanta maka dapat dilakukan perancangan kontroler dengan mencari parameter-parameter kontroler yang diperlukan. Diagram blok pada gambar 3.5 dibawah ini menunjukkan keseluruhan dari sistem perencanaan trajektori lintasan *Quadcopter* menggunakan kontroler *Neurofuzzy*.



Gambar 3.5 Diagram blok sistem keseluruhan

Pada diagram blok diatas ditunjukkan bahwa sistem ini terdiri dari beberapa subsistem atau bagian. Bagian target referensi merupakan bagian *input* berupa data koordinat lintasan yang akan dilalui *Quadcopter*. Blok referensi model merupakan blok sistem yang di inginkan dari data target referensi. Blok *error proses* (e_p) merupakan selisih dari data *output* referensi model (y_m) dengan data *output plant* (y_n). Nilai *error proses* tersebut digunakan untuk proses revisi bobot sinyal kontrol pada tahap *backward propagation* pada kontroler *Neurofuzzy*. Blok kontroler *Neurofuzzy* digunakan untuk kontrol gerak lateral *Quadcopter* pada

sumbu x dan y . Sedangkan untuk blok kontroler PD digunakan untuk stabilisasi *Quadcopter*.

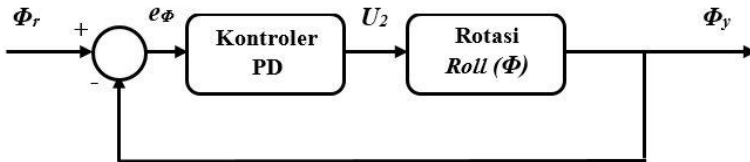
3.5. Perancangan Kontroler PD Stabilisasi Gerak Rotasi

Setelah mendapatkan model matematika dan konstanta maka dilakukan perancangan kontroler untuk telusur trajektori otomatis *Quadcopter*, menggunakan *Software Matlab*. Kontroler PD digunakan sebagai kontroler *Quadcopter* pada bagian rotasi. Gerak rotasi ini terdiri dari gerak *roll*, *pitch*, dan *yaw*. Untuk tuning parameter K_p dan τ_d pada kontroler PD digunakan prosedur tuning kontroler PD. Langkah-langkah prosedur tuning tersebut yaitu :

- Menentukan nilai K_p , dengan memberikan nilai parameter K_p sampai mendapatkan hasil mendekati *steady state* (mendekati *set point*) dan osilasi minimum
- Menentukan nilai τ_d , dengan memberikan nilai parameter τ_d sampai *overshoot* minimum

3.5.1. Perancangan Kontroler PD Stabilisasi Sudut *Roll*

Pada perancangan kontroler sudut *roll* digunakan Persamaan 3.41. Sudut *roll* berarti akan mempengaruhi pergerakan *Quadcopter* pada sumbu y . Diagram blok pengontrolan sudut *roll* ditunjukkan pada Gambar 3.6 dengan ϕ_r adalah sinyal kontrol dari kontroler translasi sumbu y . Hal tersebut dikarenakan gerak translasi dikontrol secara *cascade*.



Gambar 3.6 Diagram blok kontrol sudut *roll*

Untuk perancangan kontrol PD sudut *roll*, langkah pertama ialah mendefinisikan hubungan linear antar *variable* pada model matematika sudut *roll*. Hubungan linear tersebut terdapat pada Persamaan 3.44 - 3.46.

$$\dot{\phi} = p \quad (3.44)$$

$$\dot{\dot{\phi}} = \dot{p} \quad (3.45)$$

$$\ddot{\phi} = 0,2830qr - 15.3986q\Omega + 595.1821 U_2 \quad (3.46)$$

Menggunakan kontroler PD, berarti harus mencari nilai K_p dan τ_d . Kemudian dengan menggunakan prosedur tuning kontroler PD tersebut didapatkan nilai K_p dan τ_d untuk pengaturan sudut *roll* pada Tabel 3.4.

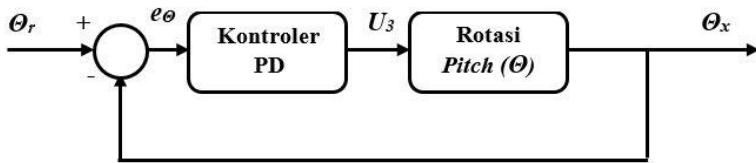
Tabel 3.4 Nilai K_p dan τ_d kontroler PD sudut *roll*

No	Parameter	Nilai
1	K_p <i>roll</i>	32
2	τ_d <i>roll</i>	2

Dari Tabel 3.4 dapat dilihat bahwa nilai K_p bernilai besar. Hal ini dikarenakan dibutuhkan respon yang cepat untuk *Quadcopter*. Respon rotasi harus lebih cepat dibanding respon translasi. Dikarenakan nilai K_p yang besar, selain respon yang semakin cepat, maka *overshot* yang dihasilkan juga semakin besar. Untuk mengantisipasi hal tersebut maka nilai τ_d juga harus besar.

3.5.2. Perancangan Kontroler PD Stabilisasi Sudut *Pitch*

Pada perancangan kontroler sudut *pitch* digunakan Persamaan 3.42. Sudut *pitch* berarti akan mempengaruhi pergerakan *Quadcopter* pada sumbu x . Diagram blok pengontrolan sudut *pitch* ditunjukkan pada Gambar 3.7 dengan θ_r adalah sinyal kontrol dari kontroler translasi sumbu x . Hal tersebut dikarenakan gerak translasi dikontrol secara *cascade*.



Gambar 3.7 Diagram blok kontrol sudut *pitch*

Hubungan antar *variable* pada model matematika gerak rotasi sudut *pitch* yang terdapat pada Persamaan 3.47-3.49.

$$\dot{\theta} = q \quad (3.47)$$

$$\ddot{\theta} = \dot{q} \quad (3.48)$$

$$\ddot{\theta} = 0,2830qr + 15.3986q\Omega + 595.1821 U_3 \quad (3.49)$$

Menggunakan kontroler PD, berarti harus mencari nilai K_p dan τ_d . Kemudian dengan menggunakan prosedur tuning kontroler PD tersebut didapatkan nilai K_p dan τ_d untuk pengaturan sudut *pitch* pada Tabel 3.5.

Tabel 3.5 Nilai K_p dan τ_d kontroler PD sudut *pitch*

No	Parameter	Nilai
1	$K_p \text{ pitch}$	32
2	$\tau_d \text{ pitch}$	2

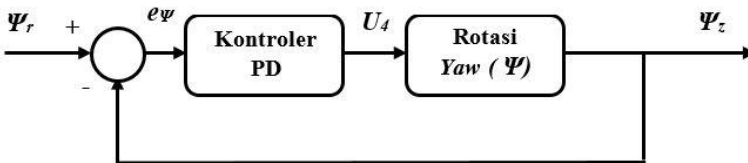
3.5.3. Perancangan Kontroler PD Stabilisasi Sudut *Yaw*

Perancangan PD pada pengendalian sudut *yaw* menggunakan Persamaan 3.43. Kontroler sudut *yaw* dirancang sebagai regulator, di mana sudut *yaw* akan dipertahankan pada sekitar 0 radian. Hubungan antar *variable* pada model matematika gerak rotasi sudut *yaw* terdapat pada Persamaan 3.50-3.52.

$$\dot{\psi} = r \quad (3.47)$$

$$\dot{\dot{\psi}} = \dot{r} \quad (3.48)$$

$$\ddot{\psi} = 0.00012pq + 830.1097 U_4 \quad (3.49)$$



Gambar 3.8 Diagram blok kontrol sudut *yaw*

Menggunakan kontroler PD, berarti harus mencari nilai K_p dan τ_d . Kemudian dengan menggunakan prosedur tuning kontroler PD tersebut didapatkan nilai K_p dan τ_d untuk pengaturan sudut *yaw* pada Tabel 3.6.

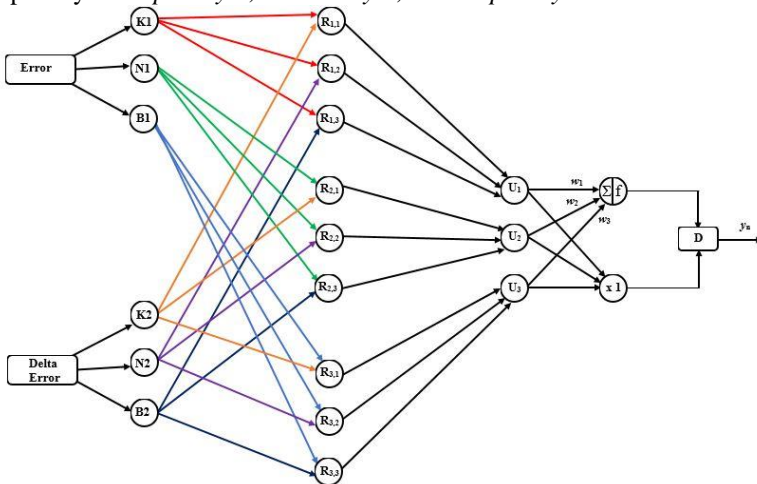
Tabel 3.6 Nilai K_p dan τ_d Kontroler PD Sudut *Yaw*

No	Parameter	Nilai
1	$K_p \text{ yaw}$	32
2	$\tau_d \text{ yaw}$	2

3.6. Perancangan Kontroler *Neurofuzzy*

Kontroler *Neurofuzzy* pada *Quadcopter* digunakan untuk navigasi melakukan trajektori lintasan dengan mengatur kecepatan motor *Quadcopter* dan mencapai target posisi yang diinginkan. Kontroler *Neurofuzzy* yang digunakan pada penelitian ini yaitu *Neurofuzzy* mamdani. Kontroler *Neurofuzzy* mamdani digunakan karena pada penelitian ini hanya diperlukan untuk perbaikan nilai tengah dari himpunan pendukung sinyal kontrol. Sedangkan untuk *rule base* tidak mengalami perbaikan.

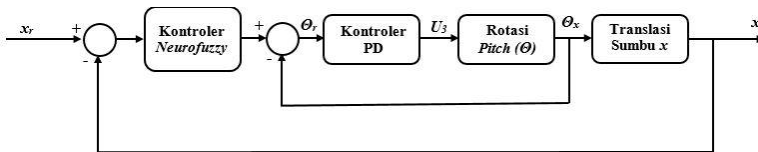
Pada gambar 3.9 menunjukkan struktur kontroler *Neurofuzzy* mamdani dengan dua *input* variabel dimana tiap *input* variabel tersebut memiliki 3 himpunan pendukung. Pada struktur tersebut terdapat 3 lapisan yaitu *input layer*, *hidden layer*, dan *output layer*.



Gambar 3.9 Struktur *Neurofuzzy* Mamdani

3.6.1. Perancangan Kontroler *Neurofuzzy* untuk sumbu x

Kontroler *Neurofuzzy* untuk trajektori lintasan pada sumbu x ini digunakan untuk menjaga agar *Quadcopter* bergerak sesuai dengan lintasannya pada sumbu x tanpa mempengaruhi perubahan sumbu y dan sumbu z . Pada gambar 3.10 merupakan blok diagram kontroler *Neurofuzzy* untuk sumbu x .



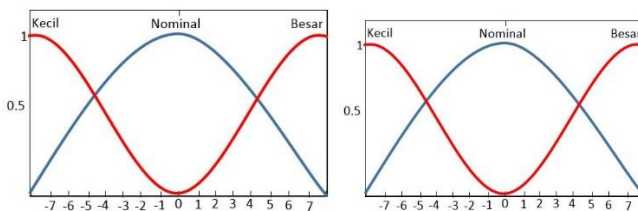
Gambar 3.10 Diagram blok kontroler *Neurofuzzy* pada sumbu x

a. Tahap *Forward Propagation*

Pada tahap *forward propagation* ini dilakukan perhitungan untuk memperoleh nilai sinyal kontrol *Neurofuzzy* yang mana sinyal kontrol tersebut digunakan untuk menggerakkan *Quadcopter*, kemudian digunakan untuk perhitungan bobot baru pada tahap *backward propagation*.

Input layer :

Pada bagian *input layer* terdapat 2 input variable seperti pada Gambar 3.18, yaitu sinyal error dan sinyal delta *error*. Kedua input variable tersebut dilakukan proses fuzzifikasi yang mana semua input tersebut memiliki 3 *membership function*. Sehingga *output* dari *input layer* ini ada sebanyak $3 \times 3 = 9$ node.



Gambar 3.11 Dua variabel dengan 3 *membership function*

Hidden layer :

Hidden layer terdiri dari 9 node yang merupakan proses fuzzifikasi dari sinyal error dan sinyal delta error. Pada lapisan ini dilakukan proses menentukan basis aturan dan rules. Untuk proses perhitungannya menggunakan persamaan 2.57, metode mamdani untuk mencari nilai minimumnya. Setelah didapat nilai minimumnya selanjutnya mencari nilai maksimum menggunakan persamaan 2.58.

Output layer :

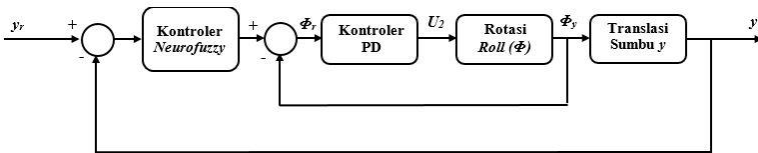
Output layer merupakan hasil perhitungan dari *hidden layer*, tahap ini melakukan proses relasi antara *hidden layer* dengan *output layer*. Pada sistem ini *output layer* ada 3 node. Setelah proses ini dilanjutkan ke proses defuzzifikasi dimana *output* dari ketiga node ini dibagi dua, ada bagian atas dan bagian bawah. Untuk perhitungannya dapat digunakan persamaan 2.59.

b. Tahap *Backward Propagation*

Pada tahap ini data *output* sinyal kontrol yang didapat dari proses *Forward propagation* digunakan untuk menggerakkan *Quadcopter* dan hasil data sinyal pergerakan tersebut dibandingkan dengan data sinyal referensi yang diberikan untuk mendapatkan nilai error seperti pada Gambar 3.10. Setelah itu dilakukan proses revisi bobot sinyal kontrol. Untuk perhitungan nilai bobot baru menggunakan persamaan 2.60.

3.6.2. Perancangan Kontroler *Neurofuzzy* untuk sumbu *y*

Kontroler *Neurofuzzy* untuk trajektori lintasan pada sumbu *y* ini digunakan untuk menjaga agar *Quadcopter* bergerak sesuai dengan lintasannya pada sumbu *y* tanpa mempengaruhi perubahan sumbu *x* dan sumbu *z*. Pada gambar 3.12 merupakan blok diagram kontroler *Neurofuzzy* untuk sumbu *y*.



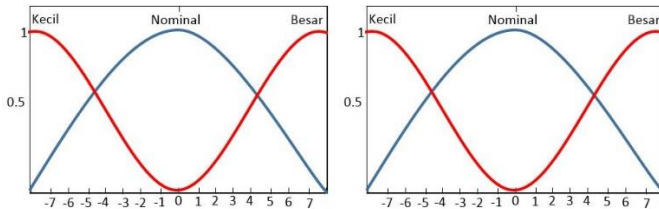
Gambar 3.12 Diagram blok kontroler *Neurofuzzy* pada sumbu *y*

a. Tahap *Forward Propagation*

Pada tahap *forward propagation* ini dilakukan perhitungan untuk memperoleh nilai sinyal kontrol *Neurofuzzy* yang mana sinyal kontrol tersebut digunakan untuk menggerakkan *Quadcopter*, kemudian digunakan untuk perhitungan bobot baru pada tahap *backward propagation*.

Input layer :

Pada bagian *input layer* terdapat 2 input variable seperti pada Gambar 3.18, yaitu sinyal error dan sinyal delta error. Kedua input variable tersebut dilakukan proses fuzzifikasi yang mana semua input tersebut memiliki 3 *membership function*. Sehingga output dari *input layer* ini ada sebanyak $3 \times 3 = 9$ node.



Gambar 3.13 Dua variabel dengan 3 *membership function*

Hidden layer :

Hidden layer terdiri dari 9 node yang merupakan proses fuzzifikasi dari sinyal error dan sinyal delta error. Pada lapisan ini dilakukan proses menentukan basis aturan dan rules. Untuk proses perhitungannya menggunakan persamaan 2.57, metode mamdani untuk mencari nilai minimumnya. Setelah didapat nilai minimumnya selanjutnya mencari nilai maksimum menggunakan persamaan 2.58.

Output layer :

Output layer merupakan hasil perhitungan dari *hidden layer*, tahap ini melakukan proses relasi antara *hidden layer* dengan *output layer*. Pada sistem ini *output layer* ada 3 node. Setelah proses ini dilanjutkan ke proses defuzzifikasi dimana output dari ketiga node ini dibagi dua, ada bagian atas dan bagian bawah. Untuk perhitungannya dapat digunakan persamaan 2.59.

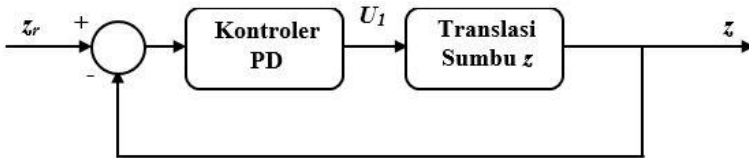
b. Tahap *Backward Propagation*

Pada tahap ini data output sinyal kontrol yang didapat dari proses *Forward propagation* digunakan untuk menggerakkan *Quadcopter* dan hasil data sinyal pergerakan tersebut dibandingkan dengan data sinyal referensi yang diberikan untuk mendapatkan nilai error seperti pada Gambar 3.10. Setelah itu

dilakukan proses revisi bobot sinyal kontrol. Untuk perhitungan nilai bobot baru menggunakan persamaan 2.60.

3.6.3. Perancangan Kontroler Pada Sumbu z

Pada perancangan kontroler untuk sumbu z ini tidak memerlukan koreksi gerak rotasi. Dalam penelitian ini kontrol gerak translasi pada sumbu z menggunakan kontroler PD. Hal ini dikarenakan *heading* dari *Quadcopter* tidak termasuk dalam bagian yang akan dikontrol, sehingga sudut gerak rotasi pada sumbu z (sudut *yaw*) sama dengan nol rad/s. Blok diagram kontroler PD pada sumbu z ditunjukkan pada gambar 3.14.



Gambar 3.14 Diagram blok kontrol gerak translasi pada sumbu z

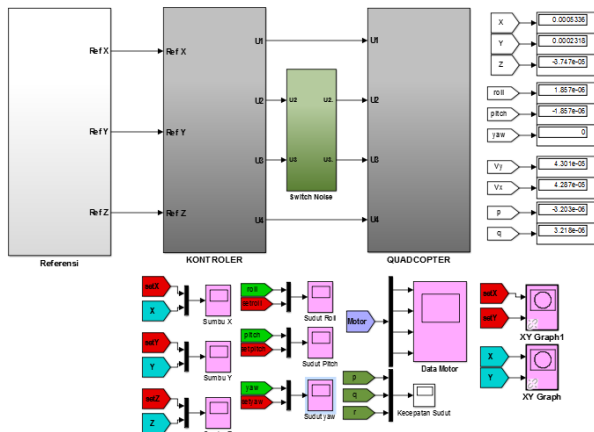
--halaman ini sengaja dikosongkan--

BAB IV HASIL PENGUJIAN DAN ANALISIS SISTEM

Pada bab ini akan dibahas mengenai pengujian dan analisa hasil rancangan sistem yang telah dibuat di bab tiga. Pengujian ini bertujuan untuk mengetahui performansi sistem dan kontroler yang telah dirancang. Kemudian dilakukan analisa terhadap metode yang digunakan. Stabilisasi sudut gerak *Quadcopter* dikontrol menggunakan kontroler PD dan untuk telusur trajektori posisi x,y dikontrol menggunakan kontroler *Neurofuzzy*.

4.1. Simulasi Pergerakan *Quadcopter*

Perancangan kontroler untuk telusur trajektori otomatis *Quadcopter* dirancang dengan sistem pengaturan *Neurofuzzy* menggunakan *software Matlab*. Pada simulasi ini *Quadcopter* bergerak sesuai dengan skenario yang diinginkan yaitu *take-off* pada ketinggian 4 meter, kemudian melakukan gerak lateral membentuk trapesium, dan *landing*. Gerak *Quadcopter* tersebut dikontrol menggunakan kontroler *Neurofuzzy* dan kontroler PD untuk stabilisasi gerak *roll*, *pitch*, dan *yaw*. Pengujian yang dilakukan meliputi pengujian tanpa *noise* dan pengujian menggunakan *noise* dengan 3 perubahan nilai varian *noise* yang berbeda. Hasil dari perancangan kontroler *Quadcopter* ini dapat dilihat pada Gambar 4.1. Pengujian tanpa *noise* dan pengujian menggunakan *noise* dilakukan dengan memindah *switch* manual pada simulasi.



Gambar 4.1 Simulasi perancangan sistem telusur trajektori *Quadcopter*

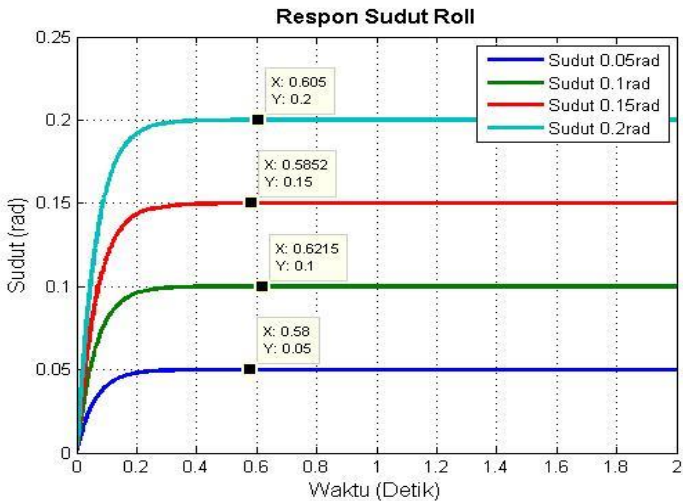
4.2. Simulasi Stabilisasi Gerak *Quadcopter*

Gerak translasi pada sumbu *x* dan sumbu *y* sangat dipengaruhi oleh perubahan gerak rotasi sudut *roll*, *pitch*, dan *yaw*. *Quadcopter* bergerak dengan trajektori koordinat (*x,y,z*) dari posisi awal pada (0 0 4) menuju (2 2 4), kemudian dilanjutkan ke posisi (-1 2 4), selanjutnya ke posisi (-2 0 4), setelah itu bergerak lagi ke posisi (-2 -2 4), dan kembali ke posisi awal ke (0 0 5) untuk melakukan pendaratan. Pada Penelitian ini nilai stabilisasi *yaw* diberi masukkan 0 sehingga orientasi *Quadcopter* (*heading*) berada pada koordinat *x'*.

Stabilisasi gerak *roll*, *pitch* dan *yaw* sangatlah penting untuk mendapatkan pergerakan *Quadcopter* seperti yang diharapkan. Hasil perancangan kontroler PD pada stabilisasi gerak *roll*, *pitch*, dan *yaw* dapat dilihat pada Tabel 4.1. Sedangkan untuk Gambar 4.2 sampai merupakan pengujian respon sudut dengan *set point* yang berbeda.

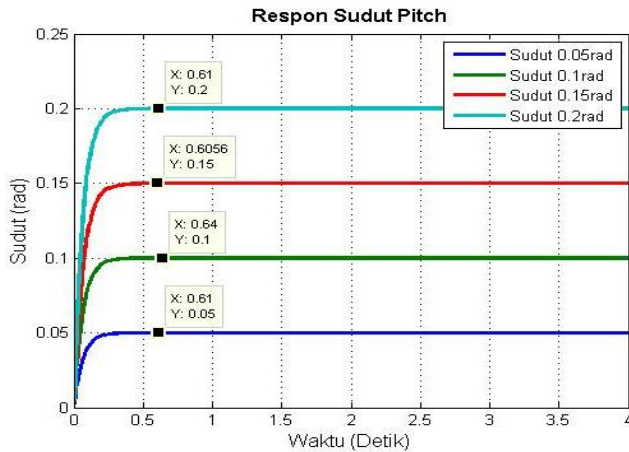
Tabel 4.1 Nilai K_p dan τ_d kontroler PD stabilisasi *Quadcopter*

Sudut	K_p	τ_d
<i>Roll</i>	32	2
<i>Pitch</i>	32	2
<i>Yaw</i>	32	2



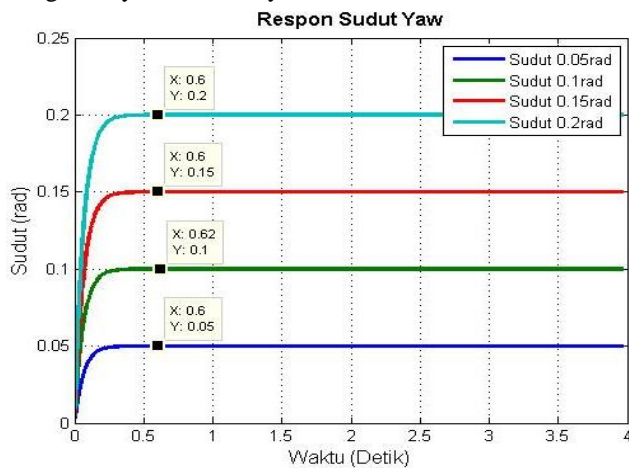
Gambar 4.2 Pengujian stabilisasi gerak *roll*

Berdasarkan grafik respon sudut *roll* pada gambar 4.2 tersebut dapat diketahui bahwa stabilisasi gerak *roll* dapat dilakukan dengan baik sesuai dengan sinyal referensinya.



Gambar 4.3 Pengujian stabilisasi gerak *pitch*

Berdasarkan grafik respon sudut *Pitch* pada gambar 4.3 tersebut dapat diketahui bahwa stabilisasi gerak *Pitch* dapat dilakukan dengan baik sesuai dengan sinyal referensinya.



Gambar 4.4 Pengujian stabilisasi gerak *yaw*

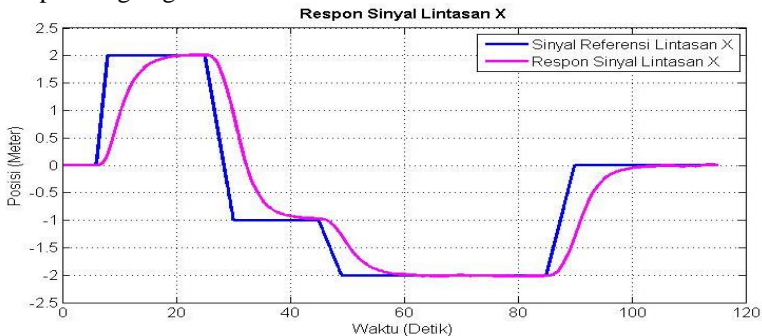
Berdasarkan grafik respon sudut *yaw* pada gambar 4.4 tersebut dapat diketahui bahwa stabilisasi gerak *yaw* dapat dilakukan dengan baik sesuai dengan sinyal referensinya. Pengujian sudut *yaw* ini dilakukan untuk menguji apakah sudut *yaw* dapat mempertahankan sudutnya pada posisi nol radian. Hal ini dikarenakan sudut *yaw* akan dijaga tetap.

4.3. Simulasi Telusur Trajektori Gerak *Quadcopter*

Setelah stabilisasi sudut *roll*, *pitch*, dan *yaw* tercapai, maka dapat diterapkan kontrol tracking pada *Quadcopter*. Pada bab tiga telah dijelaskan bahwa untuk kontrol sumbu *x* dan *y* menggunakan kontroler *Neurofuzzy*, sedangkan untuk kontrol sumbu *z* menggunakan kontroler PD. Dengan penerapan kontrol tersebut *Quadcopter* diharapkan mampu terbang dan bergerak mengikuti lintasan yang diinginkan.

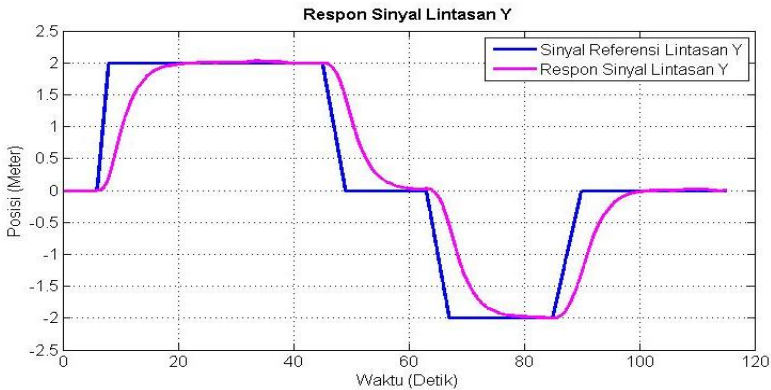
4.3.1. Telusur Trajektori Gerak Translasi *x* dan *y*

Perancangan kontroler *Neurofuzzy* ini digunakan untuk mengendalikan gerak translasi *Quadcopter*. Dalam desain logika *fuzzy*, digunakan fungsi keanggotaan *gaussian* dengan tiga himpunan pendukung. Dimana pada perancangan ini terdapat dua input yaitu sinyal *error* dan sinyal *delta error* dengan masing-masing memiliki tiga himpunan pendukung. Kontroler *neural network* pada perancangan ini digunakan sebagai penyusun struktur jaringan kontroler *fuzzy*-nya. Kemampuan kontroler *neural network* yang memiliki proses *backward propagation* untuk proses revisi bobot dimana hasil dari proses tersebut mampu merevisi bobot dari sinyal kontrol. Hasil pengujian respon pada trajektori gerak translasi sumbu *x* dan *y* dapat dilihat pada gambar 4.5 sampai dengan gambar 4.6.



Gambar 4.5 Grafik sinyal trajektori gerak translasi pada sumbu *x*

Berdasarkan hasil respon gerak translasi sumbu x pada gambar 4.5 diatas, terlihat sinyal kontrol mampu mengikuti perubahan sinyal referensi yang diberikan. Pada perubahan yang terjadi kurang dari 10 detik, memang grafik sinyal kontrol masih memiliki *error* sebesar 0,01 m dari nilai referensi yang diberikan dengan nilai RMSE 0.625.



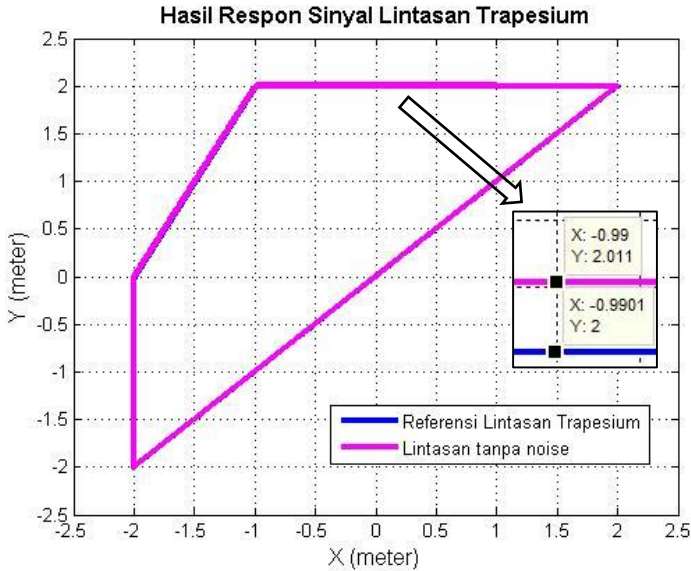
Gambar 4.6 Grafik sinyal trajektori gerak translasi pada sumbu y

Berdasarkan hasil respon gerak translasi sumbu y pada gambar 4.6 diatas, terlihat sinyal kontrol mampu mengikuti perubahan sinyal referensi yang diberikan. Pada perubahan yang terjadi kurang dari 10 detik, memang grafik sinyal kontrol masih memiliki *error* sebesar 0,01 m dari nilai referensi yang diberikan dengan nilai RMSE 0.596.

Hasil dari telusur trajektori tanpa *noise* ini dapat dilihat pada gambar 4.7, dimana hasil dari pergerakan lintasan x dan y sudah sesuai dengan referensi koordinat yang diberikan. Tabel 4.2 merupakan hasil dari perhitungan RMSE sumbu x dan y .

Tabel 4.2 Hasil *error* RMSE

Sumbu	Error	RMSE
x	0.01 m	0.625
y	0.01 m	0.596



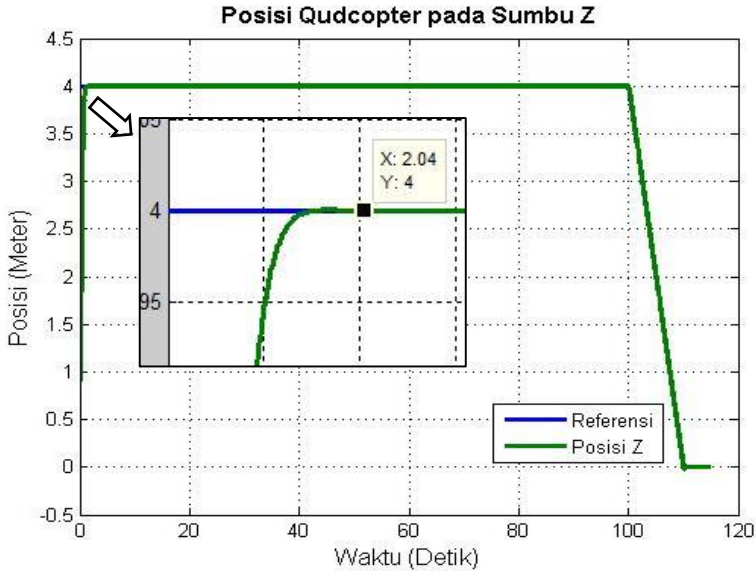
Gambar 4.7 Hasil telusur trajektori *Quadcopter* tanpa *noise*

4.3.2. Altitude sumbu *z*

Pada perancangan kontrol ketinggian pada sumbu *z* ini, *Quadcopter* diatur agar terbang pada ketinggian 4 meter diatas permukaan dan mempertahankan ketinggian pada saat bergerak menelusuri sumbu *x* maupun sumbu *y*. Tabel 4.3 merupakan nilai parameter kontrol dari kontroler PD. Parameter kontroler PD tersebut dirancang agar ketinggian *z* memiliki respon dengan waktu *steady state* 1 detik, namun pada hasil simulasi posisi *z* memiliki waktu *steady state* 2 detik dengan *error* (*Ess*) 0% seperti pada Gambar 4.8. Hal ini menunjukkan bahwa respon memiliki keterlambatan waktu untuk mencapai nilai referensi.

Tabel 4.3 Nilai K_p dan τ_d kontroler PD untuk sumbu *z*

No	Parameter	Nilai
1	$K_p \text{ thrust}$	35
2	$\tau_d \text{ thrust}$	12



Gambar 4.8 Grafik sinyal posisi sumbu z

Dari hasil simulasi tersebut dapat dilihat bahwa posisi z bergerak dari ketinggian 0 meter menuju referensi 4 meter dengan waktu *steady state* 2 detik. Setelah ketinggian tercapai kontroler akan mempertahankan ketinggian sampai akhirnya *Quadcopter* mendarat. Kontroler yang dirancang mampu mempertahankan posisi z dengan *Quadcopter* dapat bergerak pada sumbu x dan y, hal ini terlihat bahwa dari detik ke 3 sampai ke 100 posisi z tetap berada pada ketinggian 4 meter sebelum kemudian turun ke posisi 0 m.

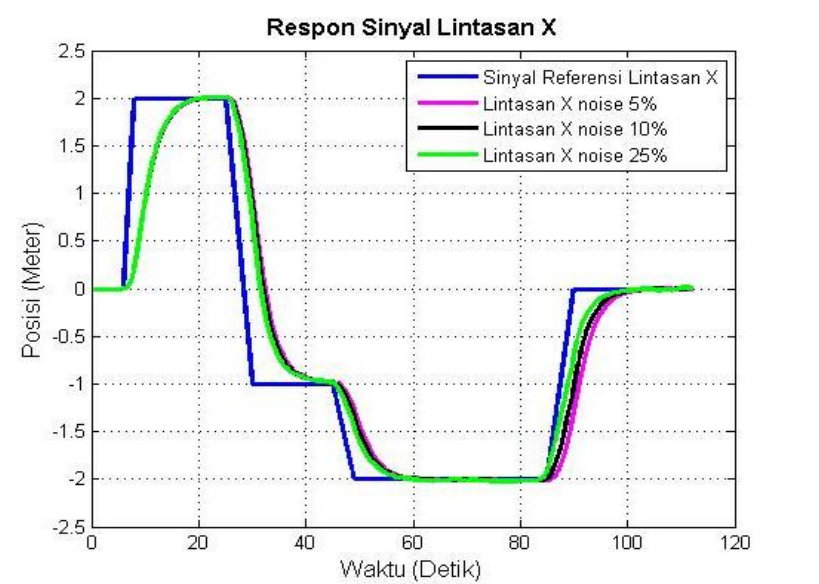
4.3.3. Telusur Trajektori Gerak *Quadcopter* dengan Noise

Sama halnya dengan pengujian tanpa *noise*, pengujian dengan *noise* dilakukan dengan mengganggu *Quadcopter* berada pada kondisi tidak ideal dengan gangguan. Gangguan yang terjadi pada *Quadcopter* memiliki dua jenis yaitu gangguan pada aktuator dan gangguan pada pengukuran. Gangguan yang diberikan pada pengujian ini yaitu gangguan hembusan angin yang diberikan pada *plant Quadcopter* pada sumbu x dan sumbu y. Pengujian ini bermaksud untuk mengetahui performansi dari kontroler *Neurofuzzy* yang telah dibuat saat terjadi gangguan pada *Quadcopter* saat bermanuver. Pengujian dilakukan sebanyak 2 kali

dengan parameter nilai gangguan berbeda. Perancangan pengujian dengan *noise* ini dilakukan dengan menambah blok fungsi *random number* pada *simulink* dengan perubahan nilai gangguan sebesar 5% (0.002), 10% (0.004), dan 25% (0.01) dari sinyal kontrol. Tabel 4.4 merupakan parameter gangguan yang akan diberikan. Hasil pengujian respon dengan gangguan pada trajektori gerak translasi sumbu *x* dan *y* dapat dilihat pada gambar 4.7 sampai dengan gambar 4.8.

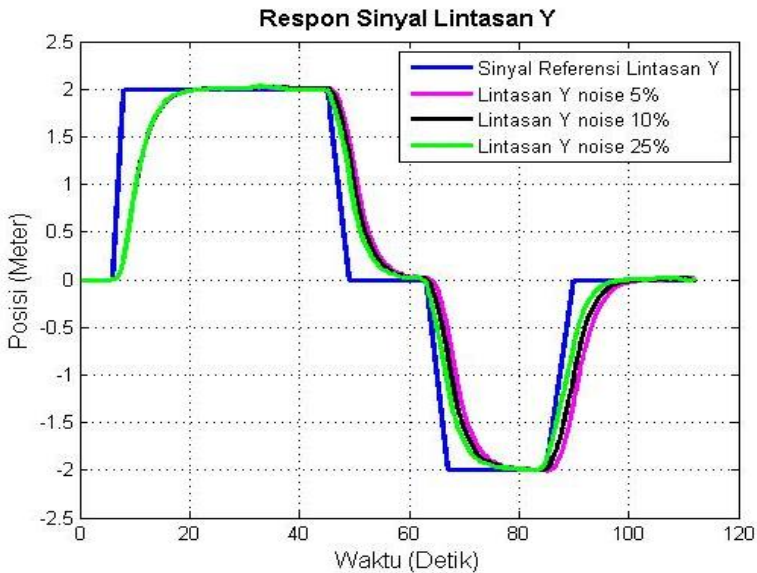
Tabel 4.4 Nilai gangguan yang diberikan

Sumbu	Noise	Persentase Noise
<i>x</i>	0.002	5%
	0.004	10%
	0.01	25%
<i>y</i>	0.002	5%
	0.004	10%
	0.01	25%



Gambar 4.9 Grafik sinyal trajektori gerak translasi pada sumbu *x* dengan *noise*

Berdasarkan hasil respon gerak translasi dengan *noise* pada sumbu *x* pada gambar 4.9 diatas, terlihat sinyal kontrol masih mampu mengikuti perubahan sinyal referensi yang diberikan walaupun terdapat *noise*. Hasil perhitungan *error* RMSE dapat dilihat pada Tabel 4.5.



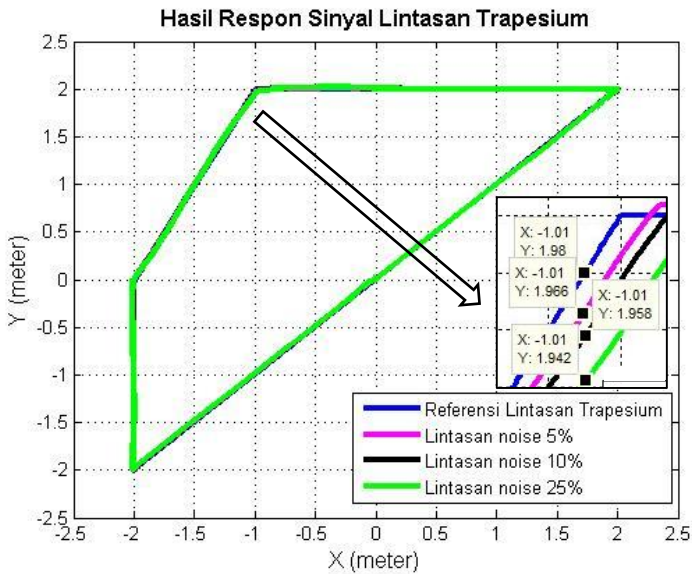
Gambar 4.10 Grafik sinyal trajektori gerak translasi pada sumbu *y* dengan *noise*

Tabel 4.5 Hasil simulasi gerak *Quadcopter* dengan variasi *noise*

No	Besar Noise	RMSE <i>x</i>	RMSE <i>y</i>
1	5%	0.628	0.601
2	10%	0.635	0.608
3	25%	0.638	0.613

Berdasarkan hasil respon gerak translasi dengan *noise* pada sumbu *y* pada gambar 4.10 diatas, terlihat sinyal kontrol masih mampu mengikuti perubahan sinyal referensi yang diberikan walaupun terdapat *noise*. Hasil perhitungan *error* RMSE dapat dilihat pada Tabel 4.5. Hasil dari telusur

trajektori dengan *noise* ini dapat dilihat pada gambar 4.11. pada hasil telusur trajektori lintasan trapesium dengan *noise* terlihat terdapat *error* lintasan ± 0.05 m, namun masih tetap sesuai dengan lintasan yang direncanakan.



Gambar 4.11 Hasil telusur trajektori *Quadcopter* dengan *noise*

BAB V

PENUTUP

Hasil dari perancangan dan penelitian Tugas Akhir dirangkum dan dirumuskan kesimpulan. Kesimpulan ini menerangkan hasil dari pengujian dan simulasi yang telah dilaksanakan.

Selama proses perancangan dan penelitian, terdapat banyak kendala yang dihadapi. Kendala tersebut telah penulis rangkum dan dirumuskan dalam bentuk saran untuk penyempurnaan dan penelitian lebih lanjut.

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis, dapat diperoleh kesimpulan sebagai berikut:

- a. Simulasi pergerakan telusur trajektori quadcopter pada lintasan trapesium dilakukan dengan waktu ± 120 detik.
- b. Pengujian tanpa noise pada kontroler *Neurofuzzy* menghasilkan sinyal respon yang baik dengan *error* lintasan x dan y sekitar ± 0.01 m dengan RMSE 0.625 dari sumbu x dan 0.596 dari sumbu y
- c. Pengujian dengan *noise* varian sekitar 5% sampai dengan 25% dari sinyal kontrol pada gerak *roll* dan *pitch* pada kontroler *Neurofuzzy* menghasilkan sinyal respon yang masih dapat mempertahankan kondisinya dengan *error* lintasan x dan y sekitar ± 0.05 m dengan RMSE terbesar pada saat gangguan 25% dengan nilai 0.638 dari sumbu x dan 0.613 dari sumbu y

5.2 Saran

Dari hasil penelitian yang dilakukan, untuk pengembangan berikutnya, disarankan beberapa hal berikut ini:

- a. Pemodelan dan pemahaman tentang *quadcopter* akan membantu perancangan sistem yang lebih baik dan mendapatkan model matematika yang lebih akurat.
- b. Penggunaan kontroler PD dan *neurofuzzy* masih bisa didapatkan nilai yang lebih optimal dengan mencari parameter tiap kontroler dengan metode *tuning* tertentu.
- c. Komunikasi antara MATLAB dengan kontroler diselesaikan agar dapat memudahkan ketika akan diimplementasikan.

--halaman ini sengaja dikosongkan--

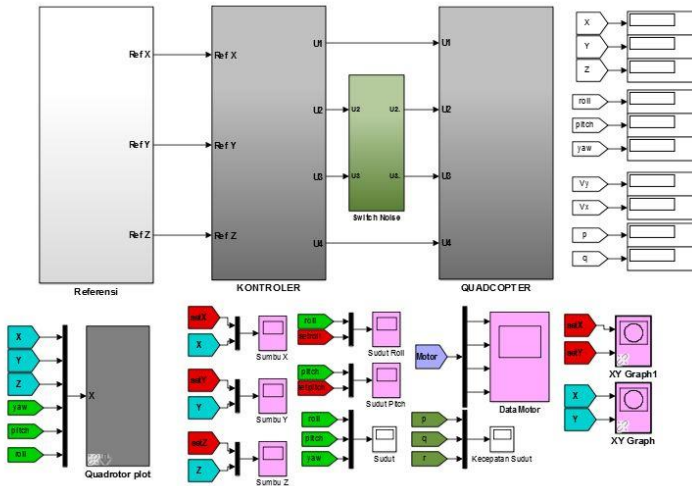
DAFTAR PUSTAKA

- [1]. Tommaso Bresciani, “*Modelling, Identification and Control of a Quadcopter Helicopter*”. Department of Automatic Control Lund University, *Thesis*, 2008
- [2]. Yuanyuan Chai, Limin Jia, Zundong Zhang, “Mamdani model based adaptive Neural Fuzzy Inference System and its Application”, *International Journal of Computer Vol3 No3*, China, 2009.
- [3]. Wira Tama, "Sistem Navigasi Waypoint Berbasis GPS (Global Positioning System) pada Quadrotor Menggunakan Kontroler PID", Tugas Akhir, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya, Juli 2011.
- [4]. Kardono, "Perancangan dan Implementasi Sistem Pengaturan Optimal LQR Untuk Menjaga Kestabilan Hover Pada Quadcopter", Tugas Akhir, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya, Juli 2012.
- [5]. Afrizal, "Implementasi Perencanaan Jalur Qbot Mobile Robot Menggunakan Metode Neuro Fuzzy", Tugas Akhir, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya, Juli 2016
- [6]. Spong, M.W, Vidyasagar, M. “*Robot Dynamic and Control*”, Prentice Hall, New Jersey, 2004
- [7]. Luukkonen, Teppo, “*Modelling and Control of Quadcopter*”, Aalto University, Espoo, 2011.
- [8]. Dinang, Sohendri, "Kontrol Tracking Fuzzy Berbasis Performa Robust Untuk Quadrotor", Tugas Akhir, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya, Juli 2016.
- [9]. Roger Jang, Chuen Tsai, Eiji Mizutani. “*Neuro-fuzzy and Soft Computing*”, Prentice Hall, Massachusetts, 1997
- [10]. Nofria, Hanafi, "Pendaratan Otomatis Quadcopter Pada Platform Yang Bergerak Menggunakan Neuro Fuzzy", Tesis, Jurusan Teknik Elektro, Program Magister, Institut Teknologi Sepuluh Nopember Surabaya, Juli 2014.
- [11]. Matilde Santos, Victoria Lopez and Francisco Morata, “Intelligent Fuzzy Controller of Quadrotor”, *Intelligent System and Knowledge Engineering International Conference. Page 141 – 146*, 2010.
- [12]. Ogata, katsuhiko “*Teknik Kontrol Automatik – terjemahan Edi Laksono*” Erlangga, Jakarta 1991.

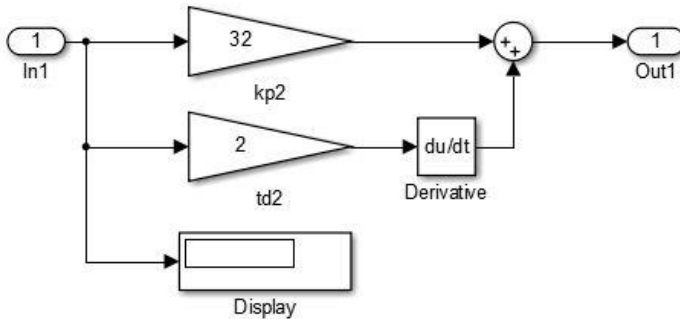
- [13]. _____. "*Turnigy Taloon Frame for Quadcopter*", www.hobbymania.co.za. Diakses pada tanggal 12 Desember 2016.
- [14]. _____. "*Ardupilot Mega 2.6 Quadcopter Flight Controller*", www.readymaderc.com. Diakses pada tanggal 12 Desember 2016.
- [15]. _____. "*Electronic Speed Controller Bulletproof 30 A quadcopter*", www.quadcopters.co.uk. Diakses pada tanggal 10 Desember 2016.
- [16]. _____. "*Sunnysky V2216 Brushless Motor 900 kV*", www.multirotorsuperstore.com. Diakses pada tanggal 24 November 2016

LAMPIRAN A

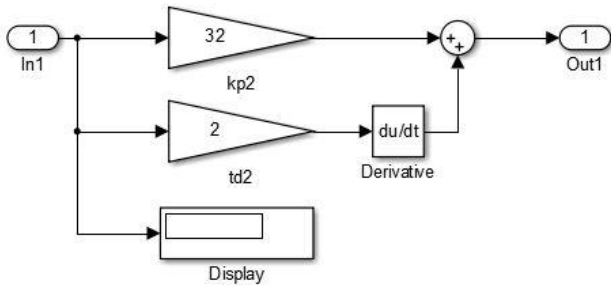
A1. Program *Simulink* Keseluruhan Sistem



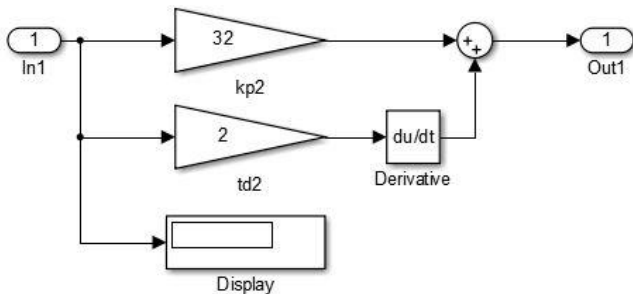
A2 Program *Simulink* Pengendalian Sudut Roll



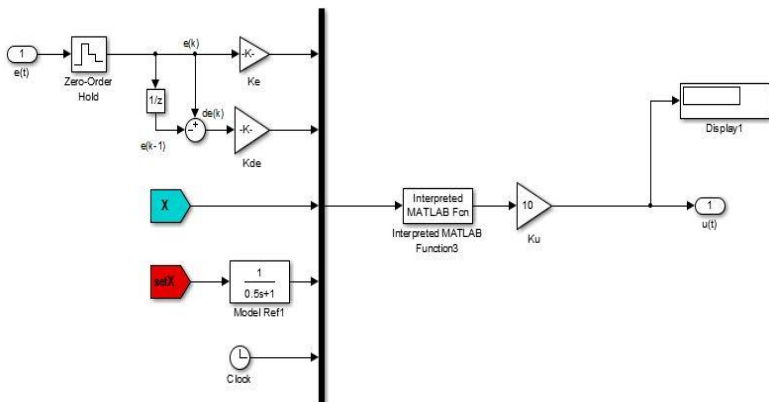
A3. Program Simulink Pengendalian Sudut Pitch



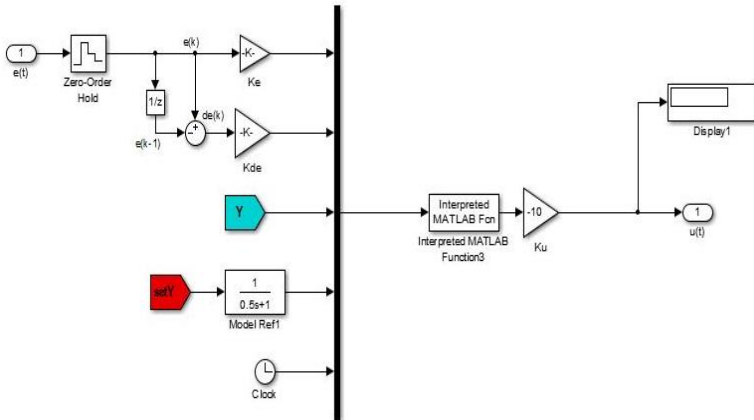
A4. Program Simulink Pengendalian Sudut Yaw



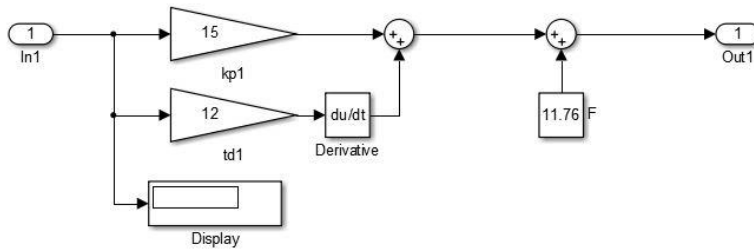
A5. Program Simulink Pengendalian Posisi X



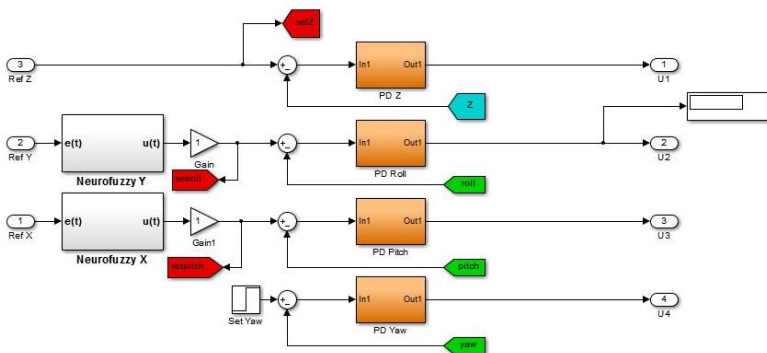
A6. Program *Simulink* Pengendalian Posisi Y



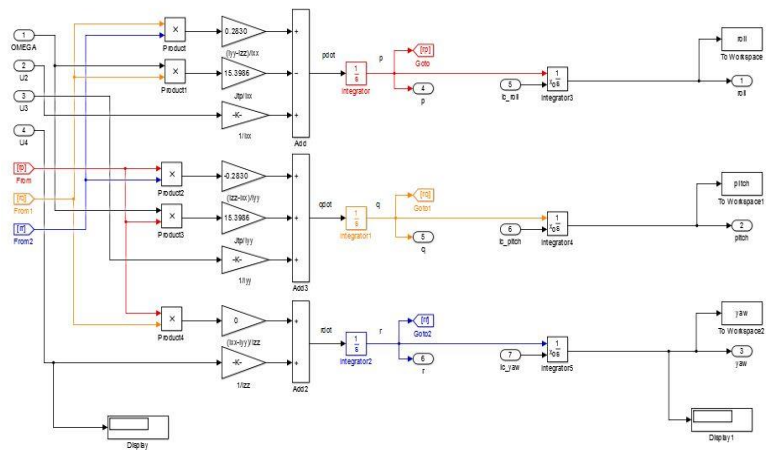
A7. Program *Simulink* Pengendalian Posisi Z



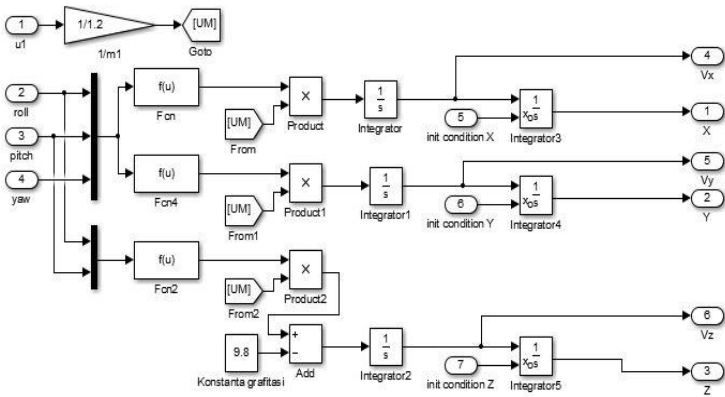
A8. Program *Simulink* Cascade Neurofuzzy Sumbu X dan Y



A9. Program Simulink Gerak Rotasi



A10. Program Simulink Gerak Translasi



LAMPIRAN B

B1. Program *Neurofuzzy* pada MATLAB

```
function neufuX = cobavarbeda_X(xx)
global ttx wx alpax lamdax

if ttx==0
    wx=[-1 0 1];
    lamdax=1;
    alpax=0.000002; %learning rate
end
%Input.....
fus_errorx=xx(1);
fus_derrorx=xx(2);
yox=xx(3);
ymx=xx(4);
ttx=xx(5);
%.....
for epoh=1:1
    %#-----FUZZIFIKASI-----
    %FUZZIFIKASI ERROR
    xf_ex=[0 0 0]';
    px=[-7.625 0 7.625];
    bx=[3.8125 3.8125 3.8125];
    for lx=1:3
        xf_ex(lx)=exp(-((fus_errorx-
px(lx))/bx(lx))^2));
    end
    %FUZZIFIKASI DELTA ERROR
    xf_dex=[0 0 0]';
    px=[-7.625 0 7.625];
    bx=[3.8125 3.8125 3.8125];
    for hx=1:3
        xf_dex(hx)=exp(-((fus_derrorx-
px(hx))/bx(hx))^2));
    end
    %-----
    %#-----RULE BASE-----
    ufx=[0 0 0]';
    rbfx=[1 1 2
```

```

        1 2 3
        2 3 3];
for ix=1:3
    for jx=1:3
        kx=rbfx(ix,jx);
        % inference rule mamdani
        ufx(kx)=max( ufx(kx),
min(xf_ex(jx),xf_dex(ix)) );
    end
end
%-----
%#-----DEFUSIFIKASI-----

atasx=wx(1)*ufx(1)+wx(2)*ufx(2)+wx(3)*ufx(3);
bawahx=ufx(1)+ufx(2)+ufx(3);
neufuX=atasx/bawahx;
% revisi bobot dari layer hiddent ke layer
output
emx=ymx-yox;
eex=1*emx;
if -2<=eex<=2
    for m=1:3
        if wx(1)<=-1.02
            break;
        else

wx(1)=wx(1)+alpax*eex*lamdax*ufx(1);
        end
        if -0.1<=wx(2)<=0.1

wx(2)=wx(2)+alpax*eex*lamdax*ufx(2);
        else
            break;
        end
        if wx(3)>=1.02
            break;
        else

wx(3)=wx(3)+alpax*eex*lamdax*ufx(3);
        end
    end
end

```

```

else
    wx(1)=wx(1);
    wx(2)=wx(2);
    wx(3)=wx(3);
end
%wx
end
end

```

B2. Program *Convert U to Omega* pada MATLAB

```

function out = utomega(in)
%u to omega
u1=in(1);
u2=in(2);
u3=in(3);
u4=in(4);

%data parameter quadcopter
b=2.2478e-6;
d=2.5e-7;
l=0.233;
%g=9.81;

omega1=sqrt((1/(4*b))*u1-(1/(2*b*l))*u3-
(1/(4*d))*u4);
omega2=sqrt((1/(4*b))*u1-
(1/(2*b*l))*u2+(1/(4*d))*u4);
omega3=sqrt((1/(4*b))*u1+(1/(2*b*l))*u3-
(1/(4*d))*u4);
omega4=sqrt((1/(4*b))*u1+(1/(2*b*l))*u2+(1/(4*d)
)*u4);

out(1)=omega1;
out(2)=omega2;
out(3)=omega3;
out(4)=omega4;

```

--halaman ini sengaja dikosongkan--

RIWAYAT PENULIS



Penulis bernama lengkap Hanief Sasongko, dilahirkan di Kota Surabaya, 17 Maret 1991. Pria yang akrab disapa Hanief ini menempuh pendidikan pertama di TK Varuna Surabaya dan melanjutkan ke sekolah dasar SDN Mojo VIII Surabaya. Lulus dari sekolah dasar tahun 2003, penulis melanjutkan pendidikan ke SMP IPIEMS Surabaya. Pada tahun 2006 dinyatakan lulus dari sekolah menengah pertama dan langsung melanjutkan ke SMA

Negeri 7 Surabaya. Setelah dinyatakan lulus pada tahun 2009, penulis melanjutkan studinya di PIKTI-ITS Surabaya selama satu tahun untuk menempuh pendidikan Informatika. Pada tahun 2010 penulis melanjutkan studinya di D3 Teknik Elektro Institut Teknologi Sepuluh Nopember dan lulus pada tahun 2013. Selanjutnya memutuskan melanjutkan ke jenjang Sarjana Jurusan Teknik Elektro ITS pada tahun 2014 dan memilih bidang studi Teknik Sistem Pengaturan. Pada bulan Januari 2017 penulis mengikuti seminar dan ujian tugas akhir sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Elektro dari Institut Teknologi Sepuluh Nopember Surabaya.

--halaman ini sengaja dikosongkan--